

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
in this Office.

出 願 年 月 日
Date of Application:

2000年 6月30日

願 番 号
Application Number:

特願2000-203063

願 人
Applicant(s):

株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

Application No. 09/761,853

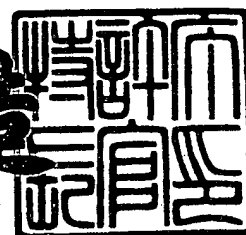
MATTINGLY, STANGER & MALUR, P.C.
1800 DIAGONAL ROAD
SUITE 370
ALEXANDRIA, VIRGINIA 22314
703-684-1120

BEST AVAILABLE COPY

2001年 1月12日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 K99009701

【提出日】 平成12年 6月30日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/00

【請求項の数】 15

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内

 【氏名】 蟹江 幸恵

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内

 【氏名】 高橋 亨

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション開発本部内

 【氏名】 青山 ゆき

【発明者】

 【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

 【氏名】 伊藤 泰樹

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社日立製作所

【代理人】

 【識別番号】 100075096

 【弁理士】

 【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 文書管理方法およびシステム並びにその処理プログラムを格納した記憶媒体

【特許請求の範囲】

【請求項 1】

改訂前後のバージョンに相当する 2 つの構造化文書ファイルの内容について、両方のバージョンに共通に含まれる構造および内容文字列からなる共通部分と、片方のバージョンにだけ固有な差分部分と、該共通部分と該差分部分のバージョン構造および内容文字列間の対応関係とを抽出する構造化文書差分抽出ステップと、

前記抽出された共通部分および前記抽出された差分部分が属するバージョンおよび双方のバージョン間での対応関係を識別するタグを付けて文書を生成する文書生成ステップと、

改訂バージョンの作成時に、該生成文書中で最新バージョンとしてタグ付けされているデータを比較対象とした前記構造化文書差分抽出ステップを実行し、該差分抽出結果に基づいて改訂バージョンの内容を上記と同様にタグ付けして該マルチバージョン文書に足し込むマルチバージョン文書更新ステップと、

前記マルチバージョン文書生成ステップおよびマルチバージョン文書更新ステップを実行することにより生成される複数バージョンの内容を含んだマルチバージョン文書をもとに、複数バージョンにおける変更箇所の一括表示および変更箇所に関する変更履歴表示を行うマルチバージョン文書表示ステップを有することを特徴とする文書管理方法。

【請求項 2】

前記構造化文書差分抽出ステップは、改訂前後のバージョンに相当する 2 つの構造化文書ファイルの内容について、両方のバージョンに共通に含まれる構造および内容文字列を共通部分として、片方のバージョンだけに固有な構造および内容文字列を差分部分として抽出し、それぞれの差分部分が改訂前バージョンに対する挿入、削除および変更のいずれに該当するかという差分分類を判定するように構成したことを特徴とする請求項 1 記載の文書管理方法。

【請求項 3】

前記マルチバージョン文書生成ステップは、前記構造化文書差分抽出ステップにより抽出された、改訂前後のバージョンに関する共通部分、差分部分、および、構造および内容文字列間の対応関係に基づいて、改訂前バージョンの内容を共通部分または差分部分毎に分割し、それぞれをバージョン属性を持つバージョン記述タグにより括った正規化データを生成し、該正規化データにおいて、共通部分を括るバージョン記述タグのバージョン属性には改訂前後の両バージョンのバージョン番号を設定し、差分部分を括るバージョン記述タグのバージョン属性には改訂前バージョンのバージョン番号を設定し、また、前記構造化文書差分抽出ステップにより抽出された差分部分のうち、改訂後バージョンに属するものをバージョン記述タグで括り、そのバージョン属性として該改訂後バージョンのバージョン番号を設定した新規差分構造を生成し、これを該正規化データ中の、構造および内容文字列間の対応関係に基づいた位置に挿入することにより、前記 2 つの構造化文書ファイルの内容をそれぞれのバージョン毎に分離可能な形で表現した XML 形式のマルチバージョン文書を生成するように構成したことを特徴とする請求項 1 記載の文書管理方法。

【請求項 4】

前記マルチバージョン文書表示ステップは、マルチバージョン文書中のバージョン記述タグに関するバージョン属性値を指定するユーザ入力を受けて、該バージョン属性値を含むバージョン記述タグで括られた構造および内容文字列を表示することにより、該バージョンの内容を再現表示し、また、該バージョン属性値を含まないバージョン記述タグで括られた構造および内容文字列を差分箇所として強調表示し、さらに、それぞれの差分箇所について、それを括るバージョン記述タグのノード属性値を取得し、それと同じノード属性値を持つバージョン記述タグにより括られた構造および内容文字列のリストを生成することにより、変更箇所に関する他バージョンでの変更履歴を生成し、これをポップアップ表示するマルチバージョン文書表示ステップを有することを特徴とする請求項 1 記載の文書管理方法。

【請求項 5】

前記マルチバージョン文書生成ステップが生成するマルチバージョン文書について、文書中に埋め込まれたバージョンの内容をファイル出力する処理として、前記マルチバージョン文書中のバージョン記述タグに関するバージョン属性値を指定するユーザ入力を受けて、該バージョン属性値を含むバージョン記述タグで括られた構造および内容文字列のみ、または、該バージョン記述タグ以下の構造をファイル出力することにより、該バージョンの内容をファイル出力する特定バージョン出力ステップを有することを特徴とする請求項 3 記載の文書管理方法。

【請求項 6】

前記マルチバージョン文書更新ステップは、前記マルチバージョン文書中のバージョン記述タグのバージョン属性値が最新バージョンに一致するものについて、該バージョン記述タグ以下を比較対象バージョンとして出力するマルチバージョン文書出力ステップと、

該比較対象バージョンと新規改訂バージョンを処理対象として差分抽出処理を実行する請求項 2 記載の構造化文書差分抽出ステップと、

前記構造化文書差分抽出ステップにより抽出された共通部分、差分部分、および、該比較対象バージョンと新規改訂バージョンの構造および内容文字列間の対応関係に基づいて、該マルチバージョン文書中の該比較対象バージョンに属する構造を、そのバージョン属性値を保持したまま、共通部分または差分部分ごとに分割した正規化データを生成し、該正規化データにおいて、共通部分を括るバージョン記述タグのバージョン属性値には、新規改訂バージョンのバージョン番号を追加し、また、前記構造化文書差分抽出ステップにより抽出された差分部分のうち、新規改訂バージョンに属するものをバージョン記述タグで括り、そのバージョン属性値として新規改訂バージョンのバージョン番号を設定した新規差分構造を生成し、これを該マルチバージョン文書中の、構造および内容文字列間の対応関係に基づいた位置に挿入することにより、新規改訂バージョンの表現を追加するように構成したことを特徴とする請求項 5 記載の文書管理方法。

【請求項 7】

前記マルチバージョン文書生成ステップおよびマルチバージョン文書更新ステ

ップは、複数のバージョンが存在する構造化文書の、すべてのバージョンの内容について、連続するバージョンに共通に含まれる構造および内容文字列を前記バージョン記述タグで括り、そのバージョン属性にそれが属するバージョンのバージョン番号を列挙した共通構造と、あるバージョンに特有の部分を前記バージョン記述タグで括り、そのバージョン属性にそれが属するバージョンのバージョン番号を設定した差分構造とを、バージョン間の構造および内容文字列の対応関係に基づいた順序で1つの文書中に埋め込んだマルチバージョン文書を出力するように構成したことを特徴とする請求項3および請求項6記載の文書管理方法。

【請求項8】

前記マルチバージョン文書生成ステップおよびマルチバージョン文書更新ステップは、前記構造化文書差分抽出ステップにより抽出される差分部分に関する差分分類を、バージョン記述タグの差分分類属性として埋め込んだマルチバージョン文書を出力するように構成したことを特徴とする請求項7記載の文書管理方法。

【請求項9】

前記マルチバージョン文書生成ステップおよびマルチバージョン文書更新ステップは、前記構造化文書差分抽出ステップにより抽出される構造および内容文字列間の対応関係を、バージョン記述タグのノード属性として埋め込んだマルチバージョン文書を出力するように構成したことを特徴とする請求項7記載の文書管理方法。

【請求項10】

前記マルチバージョン文書生成ステップおよびマルチバージョン文書更新ステップが、

前記マルチバージョン文書の表現に用いるバージョン記述タグにXML namespaceを適用することにより、処理対象とする文書データとのタグ名の重複を排除し、任意の文書構造を持つ文書データをひとまとめにしたマルチバージョン文書を出力するように構成したことを特徴とする請求項7記載の文書管理方法。

【請求項11】

前記マルチバージョン文書表示ステップは、特定のバージョンにおける差分箇

所の強調表示について、その差分箇所を括るバージョン記述タグのノード属性値を取得し、それと同じノード属性値を持つバージョン記述タグにより括られた構造および内容文字列のリストを生成することにより、変更箇所に関する他バージョンでの変更履歴を生成し、これをポップアップ表示するように構成したことを特徴とする請求項 4 記載の文書管理方法。

【請求項 1 2】

前記マルチバージョン文書表示ステップは、任意の 2 つのバージョンを画面上に並べて表示し、バージョン記述タグのノード属性を利用して、バージョン間で互いに対応関係を持つ構造および内容文字列を強調表示するように構成したことを特徴とする請求項 4 記載の文書管理方法。

【請求項 1 3】

バージョンが異なる複数の部分文書を含む文書を表示する際に、前記複数の文書部分の識別情報と該部分文書が属するバージョンを識別するタグが付けられた前記文書から最新バージョンの属性の部分文書と他のバージョンの属性の部分文書とを該属性に対応する前記識別情報に基づいて入力し、該抽出部分文書を表示することを特徴とする文書管理方法。

【請求項 1 4】

改訂前後のバージョンに相当する 2 つの構造化文書ファイルの内容について、両方のバージョンに共通に含まれる構造および内容文字列からなる共通部分と、片方のバージョンにだけ固有な差分部分と、該共通部分と該差分部分のバージョン構造および内容文字列間の対応関係とを抽出する構造化文書差分抽出手段と、

前記抽出された共通部分および前記抽出された差分部分が属するバージョンおよび双方のバージョン間での対応関係を識別するタグを付けて文書を生成する文書生成手段と、

改訂バージョンの作成時に、該生成文書中で最新バージョンとしてタグ付けされているデータを比較対象とした前記構造化文書差分抽出手段を実行し、該差分抽出結果に基づいて改訂バージョンの内容を上記と同様にタグ付けして該マルチバージョン文書に足し込むマルチバージョン文書更新手段と、

前記マルチバージョン文書生成手段およびマルチバージョン文書更新手段を実

行することにより生成される複数バージョンの内容を含んだマルチバージョン文書をもとに、複数バージョンにおける変更箇所の一括表示および変更箇所に関する変更履歴表示を行うマルチバージョン文書表示手段を有することを特徴とする文書管理システム。

【請求項 1 5】

改訂前後のバージョンに相当する 2 つの構造化文書ファイルの内容について、両方のバージョンに共通に含まれる構造および内容文字列からなる共通部分と、片方のバージョンにだけ固有な差分部分と、該共通部分と該差分部分のバージョン構造および内容文字列間の対応関係とを抽出する構造化文書差分抽出ステップと、

前記抽出された共通部分および前記抽出された差分部分が属するバージョンおよび双方のバージョン間での対応関係を識別するタグを付けて文書生成する文書生成ステップと、

改訂バージョンの作成時に、該生成文書中で最新バージョンとしてタグ付けされているデータを比較対象とした前記構造化文書差分抽出ステップを実行し、該差分抽出結果に基づいて改訂バージョンの内容を上記と同様にタグ付けして該マルチバージョン文書に足し込むマルチバージョン文書更新ステップと、

前記マルチバージョン文書生成ステップおよびマルチバージョン文書更新ステップを実行することにより生成される複数バージョンの内容を含んだマルチバージョン文書をもとに、複数バージョンにおける変更箇所の一括表示および変更箇所に関する変更履歴表示を行うマルチバージョン文書表示ステップを有する文書管理プログラムを格納したことを特徴とする記録媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、文書のバージョン管理およびバージョン間の変更箇所を差分表示する文書管理技術に関する。

【 0 0 0 2 】

【従来の技術】

近年、製品マニュアルのように広範囲に配布・再利用される文書は、配布・再利用を容易にする目的で、SGML等の構造化文書形式で記述され、蓄積されるようになってきている。また、改訂作業が頻繁に行われる文書については、改訂の度に新たな文書バージョン（以下、バージョンと省略する。）が生成され、それらは一般に別々のファイルとして管理される。これらのバージョン間の変更内容を直感的にとらえるには、バージョンの異なる2つのファイルについて、相互の文字列間の対応関係を抽出し、対応を持たない部分を差分として示す差分表示が有効である。

【 0 0 0 3 】

構造化文書を対象とした差分抽出・表示技術として、特開平9-319632号公報に記載されている構造化文書の版管理方法は、構造化文書形式で記述された2つのバージョンの一方を基準バージョンとし、2つのバージョン間の差分情報（変更箇所およびその変更内容）を抽出し、得られた差分情報を構造的に記述して基準バージョンの中に埋め込んだSGML文書（以下、これを差分埋め込み文書と呼ぶ）を出力する。これを市販のSGML文書編集ソフト等を用いて表示することにより、基準バージョンからの変更箇所および変更内容の強調表示が可能となる。また、差分埋め込み文書中に埋め込まれた変更箇所に関する構造化記述を解釈し、これに基づいて差分埋め込み文書を構造変換することにより、それぞれのバージョンを再現することもできる。すなわち、差分埋め込み文書は2つのバージョンの内容を分離可能な形で1つの文書中に効率的に表現したものと言える。

【 0 0 0 4 】

また、本公知例による版管理方法は、頻繁に改訂作業が行われる文書のバージョン管理をより効率的に行うために、あるバージョンを基準バージョンとして定め、新たなバージョンが生成される度に基準バージョンとの差分情報を抽出し、得られた差分情報のみを構造化記述した差分文書を出力し、これをバージョン管理データとして蓄積することにより、バージョン管理に要するデータ量を軽減する。この場合、それぞれの差分文書中に記述された変更箇所に関する構造化記述

を解釈し、これに基づいて基準バージョンを構造変換することにより、それぞれのバージョンを再現することが可能である。

【 0 0 0 5 】

HTMLと同様にインターネット上で扱われることを意識した構造化文書記述言語XML(eXtensible Markup Language)は、構文的にはSGMLのサブセットであり、SGMLと同様に、文書構造を自由に定義することができる。XMLで記述された文書データは、XMLに対応したWebブラウザを用いて表示・印刷できるだけでなく、特定のアプリケーションのために定義した文書構造に従って、さまざまな種類のデータを表現できるため、インターネット上でのデータ交換形式としても有効である。実際に、近年、さまざまな業界において、XMLを用いて定義された業界標準のデータ交換形式が採用されつつある。

【 0 0 0 6 】

XMLは、SGMLとは異なり、あらかじめ文書構造定義(DTD)を用意しなくても、論理構造を持った文書本体を記述することができる。その代わり、必ず、論理構造を構成する各要素を開始タグと終了タグで囲んでマーク付けしなければならない。例えば、「参加者氏名」要素を表現する場合、「参加者氏名」が「姓」と「名」の2つの要素から構成されたとすると、「<参加者氏名><姓>日立</姓><名>太郎</名></参加者氏名>」というように記述しなければならない。一方、SGMLでは、DTDを参照することにより、「参加者氏名」が「姓」と、それに続く「名」から構成されることが明白であれば、「<参加者氏名><姓>日立<名>太郎</参加者氏名>」のように、「姓」、「名」の終了タグを省略することができる。

【 0 0 0 7 】

このように、XMLでは、すべての要素を開始タグと終了タグで囲んで記述することにより、必ずしもDTDを用意しなくても論理構造を持つ文書が記述できる。この利点を生かして、さまざまな業界標準のデータを記述するためのタグセットを自由に組み合わせて、1個の文書を記述するということも可能である。XML namespaceは、1つの文書中で複数のタグセットを利用しようとした場合に生じ得る、要素名の重複の問題を回避する。例えば、数式と表を含んだ文書を作成する場合に、文書全体を記述するためのタグセットと、数式を記述するためのタグセ

ットと、表を記述するためのタグセットとの3種類のタグセットを利用したとする。仮に、文書のタグセットには文書のタイトルを表す<title>が、表のタグセットには表のタイトルを表す<title>が含まれているとすると、文書中に現れた<title>〇×調査結果</title>という記述は、どちらのタグセットに属するものかあいまいとなってしまふ。そこで、タグを記述する際に、それぞれが属するタグセットを明確にするために、例えば、表のタイトルを表すタグであれば「<hyou:title>」のように記述する。この記述において、「hyou」は表のタグセットを指定するnamespaceを示す。namespaceを用いることにより、複数タグセットのタグが含まれたXML文書であっても、アプリケーションはそれぞれのタグセットを区別して扱うことができ、また、特定のnamespaceに属するタグセットのみを処理対象とすることも可能となる。

【0008】

【発明が解決しようとする課題】

構造化文書形式で記述されたプログラマーズガイドに記載されているAPI (Application Programming Interface) を参照しながらアプリケーションプログラムの保守作業を行う場合を考えてみる。保守作業者は、保守対象プログラム中で使用されているAPIに対応するプログラマーズガイドのバージョン番号（これを旧バージョン番号と呼ぶ）をあらかじめわかっており、それ以降のバージョンで施された変更内容を把握し、それをもとにプログラムを修正するものとする。

【0009】

上記のようなアプリケーションプログラムの保守作業に際しては、旧バージョン以降変更が施されたすべてのAPIを抽出し（変更箇所）、それぞれの変更がどのバージョンにおいて行われたか（変更時期）、また、それぞれの変更箇所に関して、これまでの各バージョンでどのような変更が行われてきたか（変更履歴）、最新バージョンではどのような内容となっているか（最新状況）を調べることが必要となる。

【0010】

しかしながら、従来の方式による差分表示は、それぞれのバージョンを基準バージョンとの比較した結果を表示するものにすぎない。そのため、例えば、旧バ

ージョンを基準バージョンとして最新バージョンとの差分表示を行った場合、旧バージョンから最新バージョンまでの間に行われたすべての変更を集積した結果は得られても、それぞれのバージョンでどのような変更が行われたかに関する情報は得られない。新たな改訂バージョンを生成する度に、その直前のバージョンを基準バージョンして差分をとるやり方も考えられるが、旧バージョンと最新バージョンとの間に他バージョンが存在する場合には、他バージョンそれぞれについて、その前後のバージョンとの変更箇所および変更内容の対応関係を判断しながら、あるAPIに関する変更履歴をたどるといった、非常に手間のかかる作業になってしまう。

【0011】

また、従来の方式による差分埋め込み文書は、基準バージョンとの比較により得られる差分情報を利用して、2つのバージョンの内容を1つの文書中に効率的に表現するものであり、3つ以上のバージョンの内容を表現することはできない。

【0012】

本発明の目的は、複数のバージョンが存在する文書について、バージョンに対応する変更箇所をできるようにすることにある。

【0013】

【課題を解決するための手段】

前記目的を達成するため、本発明によるマルチバージョン文書生成方法および差分表示方法は、改訂前後のバージョンに相当する2つの構造化文書ファイルの内容について、両方のバージョンに共通に含まれる構造および内容文字列からなる共通部分と、片方のバージョンだけに固有な差分部分と、双方のバージョンの構造および内容文字列間の対応関係とを抽出する構造化文書差分抽出ステップと、抽出された共通部分および差分部分のそれぞれを、それが属するバージョンを識別するバージョン属性および双方のバージョン間での対応関係を識別するノード属性を持つバージョン記述タグによって括り、1つの文書ファイル中にまとめて記述することにより、2つのバージョンの内容を効率的に表現したマルチバージョン文書を生成するマルチバージョン文書生成ステップと、新たな改訂バージ

ョンの作成時に、該マルチバージョン文書に含まれる最新バージョンとの共通部分および差分部分、および構造および内容文字列間の対応関係を前記構造化文書差分抽出ステップにより抽出し、これに基づいて、新たな改訂バージョンの内容を上記と同様にバージョン記述タグで括り、該マルチバージョン文書中に足し込むマルチバージョン文書更新ステップを有する。

【0014】

また、本発明によるマルチバージョン文書生成方法および差分表示方法は、前記マルチバージョン文書中のバージョン記述タグに含まれるバージョン属性値を指定するユーザ入力を受けて、該バージョン属性値を含むバージョン記述タグで括られた構造および内容文字列のみを表示することにより、該バージョンの内容を再現表示し、また、該バージョン属性値を含まないバージョン記述タグで括られた構造および内容文字列を該バージョンとの差分箇所として強調表示し、さらに、それぞれの差分箇所について、それを括るバージョン記述タグのノード属性値を取得し、それと同じノード属性値を持つバージョン記述タグにより括られた構造および内容文字列のリストを生成することにより、変更箇所に関する他バージョンでの変更履歴を生成して表示するマルチバージョン文書表示ステップを有する。

【0015】

本発明によれば、文書の改訂の度に生成されるバージョン間で共通な部分を1つの文書中で共通化し、これに、それぞれのバージョンに固有な部分が足し込まれた、複数バージョンの内容を効率的に表現するXML形式のマルチバージョン文書が生成される。また、マルチバージョン文書に含まれるすべての構造および内容文字列について、それぞれが属するバージョンと、他バージョンに含まれる構造および内容文字列との対応関係が表現されるため、任意のバージョンの再現表示および差分箇所の強調表示、さらに、それぞれの差分箇所に関する他バージョンにおける構造および内容文字列を一覧表示する変更履歴表示が可能となる。

【0016】

【発明の実施の形態】

以下、本発明の第一の実施例を図面に基づいて説明する。

【 0 0 1 7 】

図 1 は本実施例のシステム構成を示す。ネットワーク 8 に接続されたコンピュータ 1 は構造化文書差分抽出機能を備えた文書管理サーバであり、コンピュータ 2 はネットワークを介して文書管理サーバ上の文書データを取得するクライアントである。本発明はサーバクライアント間の通信手段について特定するものではないが、本実施例ではインターネット上での HTTP プロトコルによる通信を前提として説明する。すなわち、文書管理サーバ 1 は HTTP サーバとしての機能を備え、クライアント 2 は Web ブラウザを介して文書管理サーバ 1 と通信するものとする。従って、文書管理サーバ 1 は、ディスプレイ 3 と、キーボード等のデータ入力装置 4 と、CPU 5 と、メモリ 6 と、SGML/HTML/XML 等で記述された構造化文書を蓄積するデータベース 7 とから構成され、メモリ 6 には、構造化文書差分抽出プログラム 6 1 と、マルチバージョン文書生成プログラム 6 2 と、マルチバージョン文書更新プログラム 6 3 と、特定バージョン出力プログラム 6 5 と、HTTP サーバプログラム 6 6 と、構造化文書差分抽出プログラム 6 1 により生成される対応関係テーブル 6 7 が保持される。また、クライアント 2 は、ディスプレイ 9 と、入力装置 1 0 と、CPU 1 1 と、メモリ 1 2 とから構成され、メモリ 1 2 には、Web ブラウザプログラム 1 2 1 と、マルチバージョン文書表示プログラム 6 4 と、特定バージョン出力プログラム 6 5 が保持される。

【 0 0 1 8 】

図 2 は本実施例の処理概要を示す。図に示すように、マルチバージョン文書の新規作成、更新、表示制御、出力制御の 4 つの処理から構成される。

【 0 0 1 9 】

マルチバージョン文書の新規作成の手順は、まず、構造化文書として記述された 2 つのバージョン（例えば、改訂前バージョン 2 0 と改訂後バージョン 2 2）について構造化文書差分抽出プログラム 6 1 を実行する。これにより、その差分抽出結果が対応関係テーブル 6 7 に格納される。次に、マルチバージョン文書生成プログラム 6 2 を実行すると、対応関係テーブル 6 7 に格納された差分抽出結果をもとに、2 つのバージョンの内容を 1 つのファイル中に表現した XML 形式のマルチバージョン文書 2 4 が生成される。

【 0 0 2 0 】

マルチバージョン文書の更新の手順は、まず、特定バージョン出力プログラム 6 5 を実行して、マルチバージョン文書 2 4 に含まれる最新バージョン 2 6 2 を出力する。次に、バージョン 2 6 2 と、追加対象バージョン（例えば、新たな改訂バージョン 3 0）を比較するために構造化文書差分抽出プログラム 6 1 を実行する。これにより、その差分抽出結果が対応関係テーブル 6 7 2 に格納される。次に、マルチバージョン文書更新プログラム 6 3 を実行すると、対応関係テーブル 6 7 2 に格納された差分抽出結果をもとに、追加対象バージョンの内容を追加したマルチバージョン文書 2 4 2 が生成される。

【 0 0 2 1 】

また、クライアント 2 における、マルチバージョン文書に関するバージョン切替え表示や対照表示等の表示制御はマルチバージョン文書表示プログラム 6 4 により、マルチバージョン文書に関する出力制御は特定バージョン出力プログラム 6 5 により行われる。

【 0 0 2 2 】

マルチバージョン文書の新規作成および更新に用いられる構造化文書の例を示す。図 2 中の改訂前バージョン 2 0、改訂後バージョン 2 2、新たな改訂バージョン 3 0 に相当する文書例として、構造化文書形式で記述されたプログラマーズガイドのバージョン V1、V2、V3 をそれぞれ図 3、図 4、図 5 に示す。図 3 および図 4 中の実線部はバージョン V1 と V2 の共通部分を、図 4 および図 5 中の点線部はバージョン V2 と V3 の共通部分を示す。また、図 3 の構造化文書データに記述されている構造を解釈することにより、図 6 に示すような木構造（これを文書木と呼ぶ。）が生成される。この文書木に含まれるすべてのノードを一意に識別するために、各ノードに対してノード ID を割り当てる。本実施例では、ルートノードのノード ID を「1」とし、その子ノード以下のノードには、「1_1」，「1_2」，…のように、親ノードのノード ID に枝番号を付加したものをノード ID として割り当てる。図 6 の文書木の各ノードに割り当てられたノード ID を、それぞれのノードの右下に示す。

【 0 0 2 3 】

以下、上記のバージョンV1とV2に関する構造化文書差分抽出プログラム61の処理手順を説明する。

【 0 0 2 4 】

まず、バージョンV1、V2のデータを差分抽出用データに変換する。本実施例では、差分抽出処理における文字列比較を容易にするため、図7のように、文書中に含まれるタグを1つの記号文字に置き換えることにする。

【 0 0 2 5 】

次に、特開平8-329079公報に記載の構造化文書差分抽出方法を用いて、上記の差分抽出用データに関する差分抽出を行う。すると、図8のような差分抽出結果が得られる。例えば、バージョンV1のノード1_3の内容文字列「int exec_diff(char* fname1, char* fname2)」は、共通部分「int exec_diff(char* 」と差分部分「fname1」と共通部分「, char* 」と差分部分「fname2」と共通部分「)」のように5つに分割、分類される。これらの分割された文字列を分割文字列と呼ぶことにする。また、各分割文字列について、それを包含するノードのノードIDに枝番号を付加したものを分割ノードIDとして割り当てることにする。例えば、上記のノード1_3に含まれる分割文字列には、その出現順に、分割ノードID「1_3_1」、「1_3_2」、「1_3_3」、「1_3_4」、「1_3_5」を割り当てる。

【 0 0 2 6 】

次に、共通部分として抽出された分割文字列の一致関係を利用して、それぞれを包含するノード間の対応関係を抽出する。例えば、上記のバージョンV1の分割ノード1_3_1「int exec_diff(char* 」と、バージョンV2の分割ノード1_3_1「int exec_diff(char* 」との一致関係により、それぞれを包含するノードであるバージョンV1のノード1_3とバージョンV2のノード1_3との対応関係が得られる。

【 0 0 2 7 】

次に、対応関係の得られたノード間における差分部分どうしの対応関係を抽出する。例えば、上記のバージョンV1のノード1_3は、バージョンV2との共通部分である分割ノード1_3_1、1_3_3と、その間に位置する差分部分である分割ノード

1_3_2「fname1」を含み、バージョンV2のノード1_3は、バージョンV1との共通部分である分割ノード1_3_1、1_3_3と、その間に位置する差分部分である分割ノード1_3_2「str1」を含む。従って、バージョンV1の分割ノード1_3_2とバージョンV2の分割ノード1_3_2は、同じ位置関係にあることがわかる。この2つの差分部分はいずれも文字「1」を含んでいるため、両者の対応関係をみとめ、これをバージョン間での変更箇所とみなすことにする。一方、バージョンV1の分割ノード1_4_1_2「、」やバージョンV2の分割ノード1_4_1_2「ます。」のように、バージョン間での対応関係が得られない場合には、それぞれを削除箇所、挿入箇所とみなす。

【 0 0 2 8 】

以上の処理により、バージョンV1について得られる差分抽出結果を格納した対応関係テーブル67-1を図9に、バージョンV2について得られる差分抽出結果を格納した対応関係テーブル67-2を図10に示す。

【 0 0 2 9 】

図11はマルチバージョン文書生成プログラムの処理フローを示す。

【 0 0 3 0 】

以下、上記のバージョンV1、V2をもとに、マルチバージョン文書を生成するものとして説明する。

【 0 0 3 1 】

まず、対応関係テーブル67-1に格納された差分抽出結果をもとに、バージョンV1の記述を正規化するデータ正規化処理を行う(6202)。正規化されたバージョンV1の内容を図13に示す。これは、構造化文書に含まれるすべてのタグおよび内容文字列が、共通部分または差分部分毎にバージョン記述タグで括られたものとなっている。バージョン記述タグとは、マルチバージョン文書にひとまとめにされるデータをバージョン毎に区別するためのタグであり、このタグで括られたデータに関するバージョン情報等を保持するための属性を持つ。本実施例では、XML namespaceを適用した「<diff:ver」で始まるタグをバージョン記述タグとする。例えば、バージョンV1の分割ノード1_4_1_1の内容文字列は、「<diff:ver vnum="1" nodeID="1_4_1_1">差分抽出処理を実行し</diff:ver>」のよう

に記述される。バージョン記述タグ中の「vnum」はバージョン属性、「nodeID」はノード属性を示す。また、図 1 3 の先頭行の「<関数一覧 xmlns:diff='http://www.xxx.yyy/diff/'>」は、「関数一覧」以下のすべての構造において、diff という namespace を適用するという namespace 宣言である。この namespace 宣言により、任意の構造を持つ構造化文書について、タグ名の重複を気にすることなく、バージョン記述タグを埋め込むことが可能となる。また、diff という namespace を知っている処理系だけが本バージョン記述タグを処理することが可能である。

【 0 0 3 2 】

次に、対応関係テーブル 6 7 - 1 中の共通部分に関する情報をもとに、正規化データ中の共通部分を括る各バージョン記述タグについて (6 2 0 4)、そのバージョン属性にバージョン番号"2"を追加し、ノード属性に対応分割ノード ID 6 7 1 2 を追加する (6 2 0 6)。例えば、上記のバージョン V1 の分割ノード 1_4_1_1 は共通部分であるため、「<diff:ver vnum="1 2" nodeID="1_4_1_1 1_4_1_1">差分抽出処理を実行し</diff:ver>」のように書き換えられる。

【 0 0 3 3 】

次に、対応関係テーブル 6 7 - 2 中の差分部分に関する情報をもとに、各差分部分を (6 2 0 8) バージョン記述タグで括り、そのバージョン属性にバージョン番号"2"を設定し、ノード属性に分割ノード ID 6 7 2 2 を設定し、差分分類属性に差分タイプ 1 2 0 9 を設定した新規差分構造を生成する (6 2 1 0)。例えば、バージョン V2 におけるノード 1_4_2 に属する差分部分「ユーザ指定のファイルに」(分割ノード 1_4_2_2) について生成される新規差分構造は「<diff:ver vnum="2" nodeID="1_4_2_2" desc="挿入">ユーザ指定ファイルに</diff:ver>」となる。

【 0 0 3 4 】

次に、上記の新規差分構造を、正規化データ中のノードの対応関係に基づいた適切な位置に挿入する (6 2 1 2)。例えば、上記のバージョン V2 におけるノード 1_4_2 に属する「ユーザ指定のファイルに」は、対応関係テーブル 6 7 - 2 中の対応ノード ID 1 2 1 2 によると、バージョン V1 における分割ノード 1_4_1_3 と 1_4_1_4 の間に挿入されたものであることがわかる。従って、正規化データ中の、

バージョン属性が"1"であり、かつ、ノード属性が1_4_1_4であるバージョン記述タグの直前に、上記の新規差分構造を挿入する。

【0035】

以上の処理により、図14に示すようなマルチバージョン文書が生成される。もし、マルチバージョン文書により、異なるバージョン間でのノードの対応関係を厳密に管理する必要がなければ、ノード属性や差分分類属性は省略してかまわない。また、さらにコンパクトな表現として、バージョンV1およびV2に共通に存在するバージョン記述タグを省略してしまってもかまわない。

【0036】

図12はデータ正規化プログラムの処理フローを示す。

【0037】

以下、バージョンV1とV2についての差分抽出結果をもとに、バージョンV1の内容を正規化するものとして説明する。

【0038】

まず、マルチバージョン文書新規作成のための処理であれば(62022)、文書中に含まれるすべてのタグおよびその内容文字列をバージョン記述タグで括り、そのバージョン属性に"1"を設定する(62024)。また、文書木のルートノードに該当するタグに、バージョン記述タグに関するnamespace宣言を挿入する(62025)。これにより、上記のバージョンV1のノード1_4_1に関する記述「<段落>差分抽出処理を実行し、抽出結果を出力します。</段落>」は「<diff:ver vnum="1"><段落><diff:ver vnum="1">差分抽出処理を実行し、抽出結果を出力します。</diff:ver></段落></diff:ver>」と、ルートノードの開始タグとなる「<関数一覧>」は「<関数一覧 xmlns:diff='http://www.xxx.yyy/diff/'>」と書き換えられる。

【0039】

次に、対応関係テーブル67-1におけるノード間の対応関係(6711と6718)を参照し、正規化対象バージョン(ここではバージョンV1)に属する各ノードについて(62026)、他方のバージョン(ここではバージョンV2)に属する2つ以上のノードが対応づけられている場合(62028)、それを括る

バージョン記述タグを含めた構造XXの属性を保持したまま、対応するノードY毎に分割する（62030）。例えば、図9に示すように、バージョンV1のノード1_4_1には、バージョンV2の2つのノード1_4_1、1_4_2が対応づけられている。従って、上記の正規化記述を対応ノードIDの1_4_1から1_4_2への変わり目で分割して、「<diff:ver vnum="1"><段落><diff:ver vnum="1">差分抽出処理を実行し、</diff:ver></段落></diff:ver><diff:ver vnum="1"><段落><diff:ver vnum="1">抽出結果を出力します。</diff:ver></段落></diff:ver>」と置き換える。

【0040】

次に、対応関係テーブル67-1中の正規化対象バージョンに関する共通部分および差分部分が（62032）、上記の正規化記述中のバージョン属性に正規化対象バージョンを含むバージョン記述タグで括られた構造Aと部分的に一致する場合には（62036）、その構造Aを括るバージョン記述タグを含めた構造Aの属性を保持したまま、一致部分とそれ以外とに分割する（62038）。例えば、上記の正規化記述に含まれる「<diff:ver vnum="1">抽出結果を出力します。</diff:ver>」は、共通部分「抽出結果を」と共通部分「出力します。」とに分かれて抽出されているため、「<diff:ver vnum="1">抽出結果を</diff:ver><diff:ver vnum="1">出力します。</diff:ver>」と置き換える。以上の処理により、図13のような正規化データが生成される。

【0041】

図17はマルチバージョン文書更新プログラムの処理フローを示す。

【0042】

以下、バージョンV1とV2から生成されたマルチバージョン文書について、新たな改訂バージョンV3の内容を追加するものとして説明する。

【0043】

あらかじめ、更新対象となるマルチバージョン文書に含まれる最新のバージョン（ここではバージョンV2）と追加対象バージョン（ここではバージョンV3）について、構造化文書差分抽出プログラムが実行され、その差分抽出結果が得られているものとする。ただし、図4に示すバージョンV2の代わりに、図15に示すような、マルチバージョン文書に埋め込まれている最新バージョン（バージョン

V2) の内容をそれを括るバージョン記述タグごとに取り出したもの（タグ付きバージョンV2a）を比較対象とする。タグ付きバージョンV2aとV3とを比較した差分抽出結果の例を図16に示す。図16において、バージョン記述タグは、原文書に含まれるタグとは区別可能な記号文字に置き換えたものとなっている。

【0044】

以下、上記の差分抽出結果に基づき、更新処理の手順を説明する。

【0045】

まず、バージョンV2aに関する対応関係テーブル中の共通部分および差分部分に関する情報をもとに、マルチバージョン文書について、バージョンV2を正規化対象としたデータ正規化処理を行う（6302）。次に、バージョンV2aに関する対応関係テーブル中の共通部分に関する情報をもとに、正規化データ中の共通部分を括るバージョン記述タグについて（6304）、そのバージョン属性にバージョン番号"3"を追加し、ノード属性に対応分割ノードIDを追加する（6306）。次に、バージョンV3に関する対応関係テーブル中の差分部分に関する情報をもとに、各差分部分を（6308）バージョン記述タグで括り、そのバージョン属性にバージョン番号"3"を設定し、ノード属性に分割ノードIDを設定し、差分類属性に差分タイプを設定した新規差分構造を生成する（6310）。次に、この新規差分構造を正規化データ中のノードの対応関係に基づいた適切な位置に挿入する（6312）。以上の処理により、図18に示すようなマルチバージョン文書が生成される。

【0046】

図19は特定バージョン出力プログラムの処理フローを示す。

【0047】

まず、出力対象とするバージョン番号の指定、およびデータ出力時におけるバージョン記述タグの出力／非出力の指定を受け付ける（6502）。次に、マルチバージョン文書中のバージョン記述タグのバージョン属性に、指定のバージョン番号が含まれているか否かチェックし（6506）、それを含むバージョン記述タグ以下の構造を上記の出力指定に基づきデータ出力する。出力指定が「出力しない」の場合はバージョン記述タグの内部に含まれる構造のみ出力し（651

0)、「出力する」の場合はそのバージョン記述タグごとファイル出力する(6512)。すなわち、上記の出力指定が「出力しない」であれば、マルチバージョン文書の生成に用いた各バージョンの原データが出力され、「出力する」であれば、タグ付きバージョンが出力される。

【0048】

図20はマルチバージョン文書表示プログラムの処理フローを示す。

【0049】

まず、表示対象バージョンおよび表示パターンの指定を受け付ける(6402)。本実施例では、表示パターンとして、1つのバージョンの内容表示を行う「バージョン指定表示」と、複数バージョンを並べた比較表示を行う「バージョン対照表示」の2通りの画面表示について説明する。「バージョン対照表示」は、各バージョンの内容をテキストベースで比較表示するテキスト対照表示と、文書木ベースで比較表示するツリー対照表示のいずれかを選択できるものとする。また、「バージョン指定表示」における表示オプションとして、例えば、バージョンV2からV3までといったバージョン範囲を指定して、そのバージョン範囲において何らかの変更を受けた箇所(差分箇所)を強調表示する差分表示、およびその差分箇所に関する変更履歴の表示を指定できるものとする。

【0050】

まず、「バージョン指定表示」が表示パターンとして指定された場合、バージョン指定表示処理(6404)により、マルチバージョン文書中のバージョン記述タグのバージョン属性を利用した指定バージョンの画面表示が行われる。

【0051】

次に、上記のバージョン表示における表示オプションとして、特定のバージョン範囲における差分表示が指定された場合、差分表示処理(6406)により、マルチバージョン文書中のバージョン記述タグのバージョン属性値の先頭要素が指定のバージョン範囲を満たすものを差分箇所として強調表示する。また、上記の差分箇所についての変更履歴表示が指定された場合、変更履歴表示処理(6408)により、対象とする差分箇所を括るバージョン記述タグのノード属性をもとに、マルチバージョン文書中の同じノード属性を持ったバージョン記述タグで

括られた文字列をリストにした変更履歴を生成し、これをポップアップ表示する。上記の差分表示および変更履歴表示を指定した画面表示の例を図 2 1 に示す。

【0052】

また、表示パターンとして、「バージョン対照表示」のテキスト対照表示が指定された場合には、テキスト対照表示処理（6410）により、マルチバージョン文書中のバージョン記述タグのバージョン属性およびノード属性をもとにノード間の対応関係を取得し、これに基づいた図 2 2 に示すような画面表示を行う。図 2 2 は、バージョン V1 の内容を表示領域 2 2 1 に表示し、バージョン V2 の内容を表示領域 2 2 2 に表示したものである。また、表示領域 2 2 1 中の「fname1」をクリックすると、それに対応する表示領域 2 2 2 中の「str1」が表示領域 3 9 2 の中央に表示されるというような表示制御も可能である。一方、ツリー対照表示が指定された場合には、ツリー対照表示処理（6412）により、図 2 2 の対照表示と同様な、図 2 3 に示すような文書木ベースでの対照表示を行う。

【0053】

以上が第一の実施例の説明である。

【0054】

本実施例によれば、文書の改訂の度に生成されるバージョン間で共通な部分を 1 つの文書中で共通化し、これに、それぞれのバージョンに固有な部分が足し込まれた、複数バージョンの内容を効率的に表現する XML 形式のマルチバージョン文書が生成されるため、バージョン管理に要するデータ量が軽減されるとともに、ネットワーク上での複数バージョンのデータ交換が容易になる。また、マルチバージョン文書に含まれるすべての構造および内容文字列について、それぞれが属するバージョンと、他バージョンに含まれる構造および内容文字列との対応関係が表現されるため、任意のバージョンの再現表示および差分箇所の強調表示、さらに、それぞれの差分箇所に関する他バージョンにおける構造および内容文字列を一覧表示する変更履歴表示が可能となり、ユーザが文書に施された変更内容および変更時期を容易に把握できるようになる。

【0055】

以下に、第二の実施例について説明する

図 2 4 に本実施例の処理概要を示す。本実施例は、文書の内容が英語で記述された英文バージョンとその和訳である和文バージョンのように、同一の文書構造を持ち、その内容の記述言語だけが異なるような複数のバージョンを 1 つの文書にまとめたマルチリンガル文書の生成を目的とするものである。同一の文書構造を持つことを利用した、それぞれのバージョンに含まれるノード間の対応づけが可能であるため、第一の実施例と違って、バージョン間の対応関係を得るための構造化文書差分抽出プログラムは必要ない。その代わり、対応関係テーブル 6 9 を生成するための対応関係テーブル作成プログラム 6 8 を有する。

【 0 0 5 6 】

マルチリンガル文書の新規作成の手順は、まず、英文バージョン 4 0 と、その和文バージョン 4 2 について、対応関係テーブル作成プログラム 6 8 を実行し、対応関係テーブル 6 9 を生成する。次に、マルチバージョン文書生成プログラム 6 2 を実行すると、対応関係テーブル 6 9 に格納された対応関係をもとに、両バージョンの内容を 1 つのファイル中に表現したマルチリンガル文書 4 4 が生成される。

【 0 0 5 7 】

英文バージョン 4 0、和文バージョン 4 2 の例をそれぞれ図 2 5、図 2 6 に示す。それぞれのバージョンの各ノードの内容文字列として含まれる文単位の対応関係を表した対応テーブル 6 9 を図 2 7、図 2 8 に示す。また、この対応関係をもとに生成されるマルチリンガル文書の例を図 2 9 に示す。ただし、バージョン記述タグには、記述言語を識別するためのバージョン種類属性 `vkind` を持たせる。図において、`vkind="e"` であるバージョン記述タグで括られた部分は英文バージョン 4 0 に属するものを、`vkind="j"` であるバージョン記述タグで括られた部分は和文バージョン 4 2 に属するものを表す。

【 0 0 5 8 】

また、上記のマルチリンガル文書に対して、上記の英文および和文バージョンと同一の文書構造を持つ、フランス語で記述された仏文バージョン 5 0 の内容を追加するマルチリンガル文書の更新の手順は、上記マルチリンガル文書中の一方のバージョンとの対応関係を格納した対応関係テーブル 6 9 2 を生成した後、マ

ルチバージョン更新処理プログラムを実行する。これにより英文、和文および仏文を含むマルチリンガル文書が生成される。

【 0 0 5 9 】

クライアント 2 は、Web ブラウザ プログラム 1 2 1 を利用して、データベース 7 に蓄積されているマルチリンガル文書を取得し、これをマルチバージョン文書と見なして、第一の実施例と同様な表示制御および出力制御を行う。

【 0 0 6 0 】

以上が第二の実施例である。

【 0 0 6 1 】

本実施例によれば、文書の改訂の度に生成されるバージョン間で共通な部分を 1 つの文書中で共通化し、これに、それぞれのバージョンに固有な部分が足し込まれた、複数バージョンの内容を効率的に表現する XML 形式のマルチバージョン文書が生成されるため、バージョン管理に要するデータ量が軽減されるとともに、ネットワーク上での複数バージョンのデータ交換が容易になる。また、マルチバージョン文書に含まれるすべての構造および内容文字列について、それぞれが属するバージョンと、他バージョンに含まれる構造および内容文字列との対応関係が表現されるため、任意のバージョンの再現表示および差分箇所の強調表示、さらに、それぞれの差分箇所に関する他バージョンにおける構造および内容文字列を一覧表示する変更履歴表示が可能となり、ユーザが文書に施された変更内容および変更時期を容易に把握できるようになる。

【 0 0 6 2 】

【発明の効果】

本発明によれば、複数のバージョンが存在する文書について、バージョンに対応する変更箇所を表示できるため、ユーザがバージョンにおける内容の違いを容易に把握できる。

【図面の簡単な説明】

【図 1】

本発明を適用したシステム構成を示す図である。

【図 2】

本発明の処理概要を示す図である。

【図 3】

本発明の処理対象となる構造化文書データ例：プログラマーズガイドV1を示す図である。

【図 4】

本発明の処理対象となる構造化文書データ例：プログラマーズガイドV2を示す図である。

【図 5】

本発明の処理対象となる構造化文書データ例：プログラマーズガイドV3を示す図である。

【図 6】

プログラマーズガイドV1の文書木を示す図である。

【図 7】

差分抽出用データ例を示す図である。

【図 8】

プログラマーズガイドV1,V2間の差分抽出結果を示す図である。

【図 9】

プログラマーズガイドV1に関する対応関係テーブルを示す図である。

【図 1 0】

プログラマーズガイドV2に関する対応関係テーブルを示す図である。

【図 1 1】

マルチバージョン文書生成プログラムのPAD図である。

【図 1 2】

データ正規化プログラムのPAD図である。

【図 1 3】

プログラマーズガイドV1の正規化データを示す図である。

【図 1 4】

プログラマーズガイドV1,V2の内容を表現したマルチバージョン文書を示す図

である。

【図 1 5】

タグつきバージョンV2aを示す図である。

【図 1 6】

タグつきバージョンV2aと、プログラマーズガイドV3間の差分抽出結果を示す図である。

【図 1 7】

マルチバージョン文書更新プログラムのPAD図である。

【図 1 8】

プログラマーズガイドV1,V2,V3の内容を表現したマルチバージョン文書データを示す図である。

【図 1 9】

特定バージョン出力プログラムのPAD図である。

【図 2 0】

マルチバージョン文書表示プログラムのPAD図である。

【図 2 1】

マルチバージョン文書に関する差分表示および変更履歴表示例を示す図である。

【図 2 2】

マルチバージョン文書に関するテキスト対照表示例を示す図である。

【図 2 3】

マルチバージョン文書に関するツリー対照表示例を示す図である。

【図 2 4】

第二の実施例における処理概要を示す図である。

【図 2 5】

第二の実施例における処理対象となる構造化文書データ例：英文バージョンを示す図である。

【図 2 6】

第二の実施例における処理対象となる構造化文書データ例：和文バージョンを

示す図である。

【図 2 7】

英文バージョンに関する対応関係テーブルを示す図である。

【図 2 8】

和文バージョンに関する対応関係テーブルを示す図である。

【図 2 9】

英文および和文バージョンの内容を表現したマルチリンガル文書を示す図である。

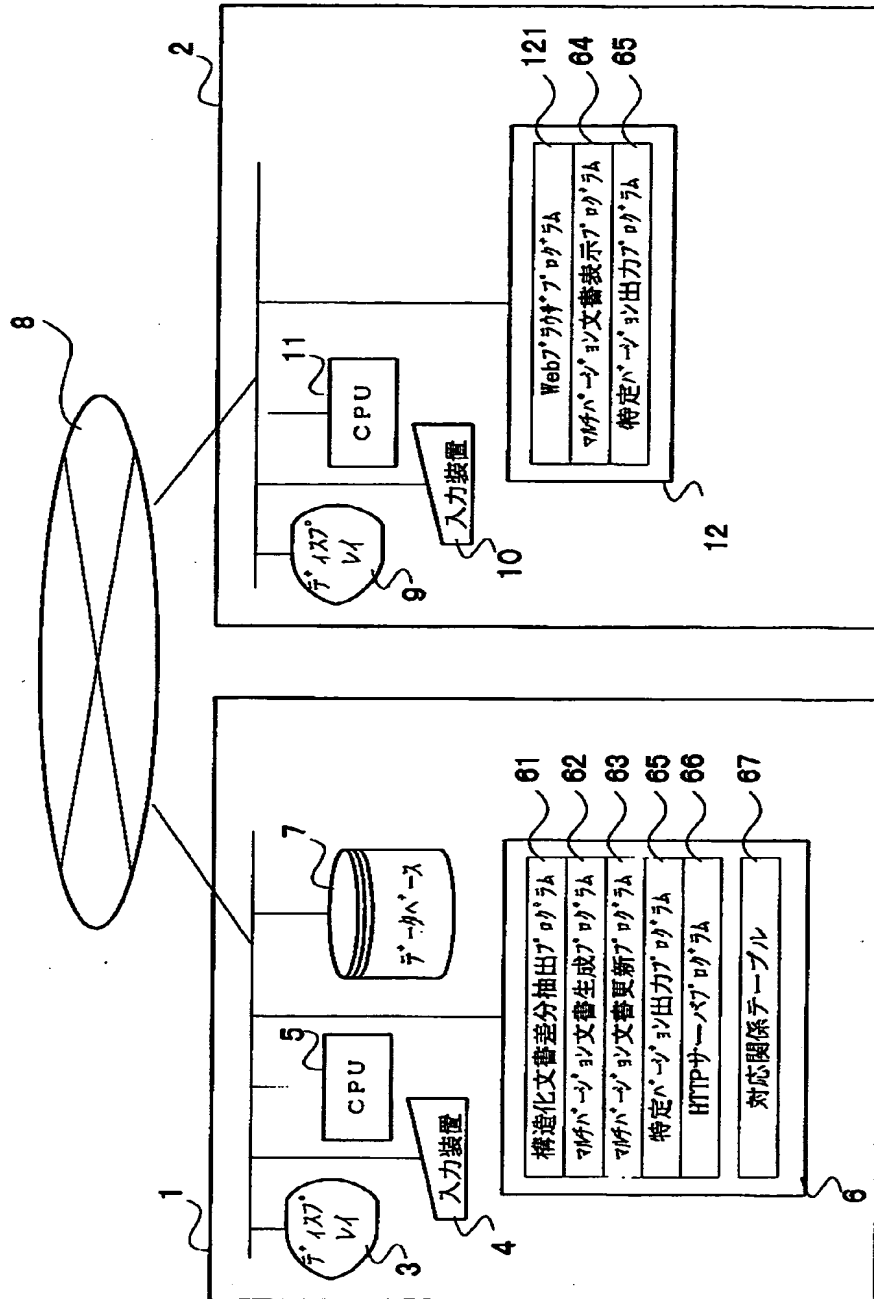
【符号の説明】

- 1 文書管理サーバ
- 2 クライアント
- 7 データベース
- 8 ネットワーク
- 2 0 改訂前バージョン
- 2 2 改訂後バージョン
- 3 0 新たな改訂バージョン
- 6 1 0 構造化文書差分抽出プログラム
- 6 2 0 マルチバージョン文書生成プログラム
- 6 3 0 マルチバージョン文書更新プログラム
- 6 4 0 マルチバージョン文書表示プログラム
- 6 5 0 特定バージョン出力プログラム
- 6 7、6 7 2 対応関係テーブル
- 2 4、2 4 2 マルチバージョン文書
- 2 6、2 6 2 マルチバージョン文書から出力されるバージョン
- 2 8 マルチバージョン文書の表示画面

【書類名】 図面

【図 1】

図 1



【図 3】

図 3

<関数一覧>

...

<関数情報>

<関数名称>exec_diff</関数名称>

<機能>差分抽出処理の実行</機能>

<形式>int exec_diff(char* fname1, char* fname2)</形式>

<説明>

<段落>差分抽出処理を実行し、抽出結果を出力します。</段落>

</説明>

<引数リスト>

<引数>×<名称>fname1</名称>×<内容>比較対象ファイル名</内容>×</引数>

<引数>×<名称>fname2</名称>×<内容>比較対象ファイル名</内容>×</引数>×</引数リスト>

<戻り値リスト>

<戻り値>×<値>0</値>×<詳細>正常終了</詳細>×</戻り値>×</戻り値リスト>

<注意事項>特になし</注意事項>

</関数情報>

...

</関数一覧>

【図 4】

図 4

```

<関数一覧>
...
<関数情報>
  <関数名称>exec_diff</関数名称>
  <機能>差分抽出処理の実行</機能>
  <形式>int exec_diff( char* str1, char* str2 )</形式>
  <説明>
    <段落>差分抽出処理を実行します。</段落>
    <段落>抽出結果をユーザ指定のファイルに出力します。</段落>
  </説明>
  <引数リスト>
    <引数×名称>str1</名称×内容>比較対象文字列</内容×/引数>
    <引数×名称>str2</名称×内容>比較対象文字列</内容×/引数×/引数リスト>
  <戻り値リスト>
    <戻り値×値>0</値×詳細>正常終了</詳細×/戻り値>
    <戻り値×値>ERR_1</値×詳細>引数が不正です。</詳細×/戻り値×/戻り値リスト>
  </関数情報>
...
</関数一覧>

```

【図5】

図5

<関数一覧>

...

<関数情報>

<関数名>exec_diff</関数名>

<機能>指定された差分抽出モードに基づいた、差分抽出処理の実行</機能>

<形式>int exec_diff(char* str1, char* str2, int flg)</形式>

<説明>

<段落>指定された差分抽出モードに基づき、差分抽出処理を実行します。</段落>

<段落>差分抽出モードには、一般モードと詳細モードがあります。</段落>

<段落>抽出結果をユーザ指定のファイルに出力します。</段落>

</説明>

<引数リスト>

<引数×名称>str1</名称×内容>比較対象文字列</内容×/引数>

<引数×名称>str2</名称×内容>比較対象文字列</内容×/引数>

<引数×名称>flg</名称×内容>差分抽出モード指定</内容×/引数×/引数リスト>

<戻り値リスト>

<戻り値×値>0</値×詳細>正常終了</詳細×/戻り値>

<戻り値×値>ERR_1</値×詳細>比較対象文字列が不正です。</詳細×/戻り値>

<戻り値×値>ERR_2</値×詳細>初期化処理が実行されていない。</詳細×戻り値>

</戻り値リスト>

<注意事項>本処理を実行する前に、init_diff0を実行すること。</注意事項>

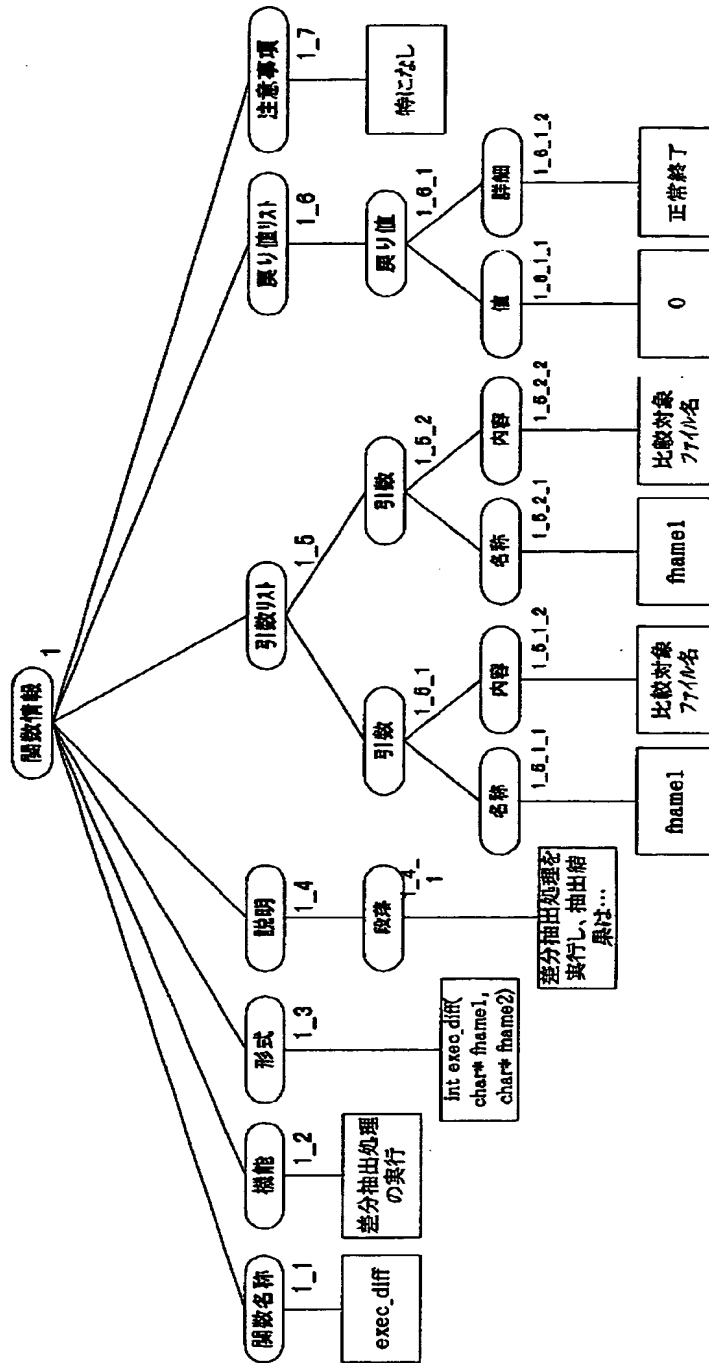
</関数情報>

...

</関数一覧>

【図 6】

図 6



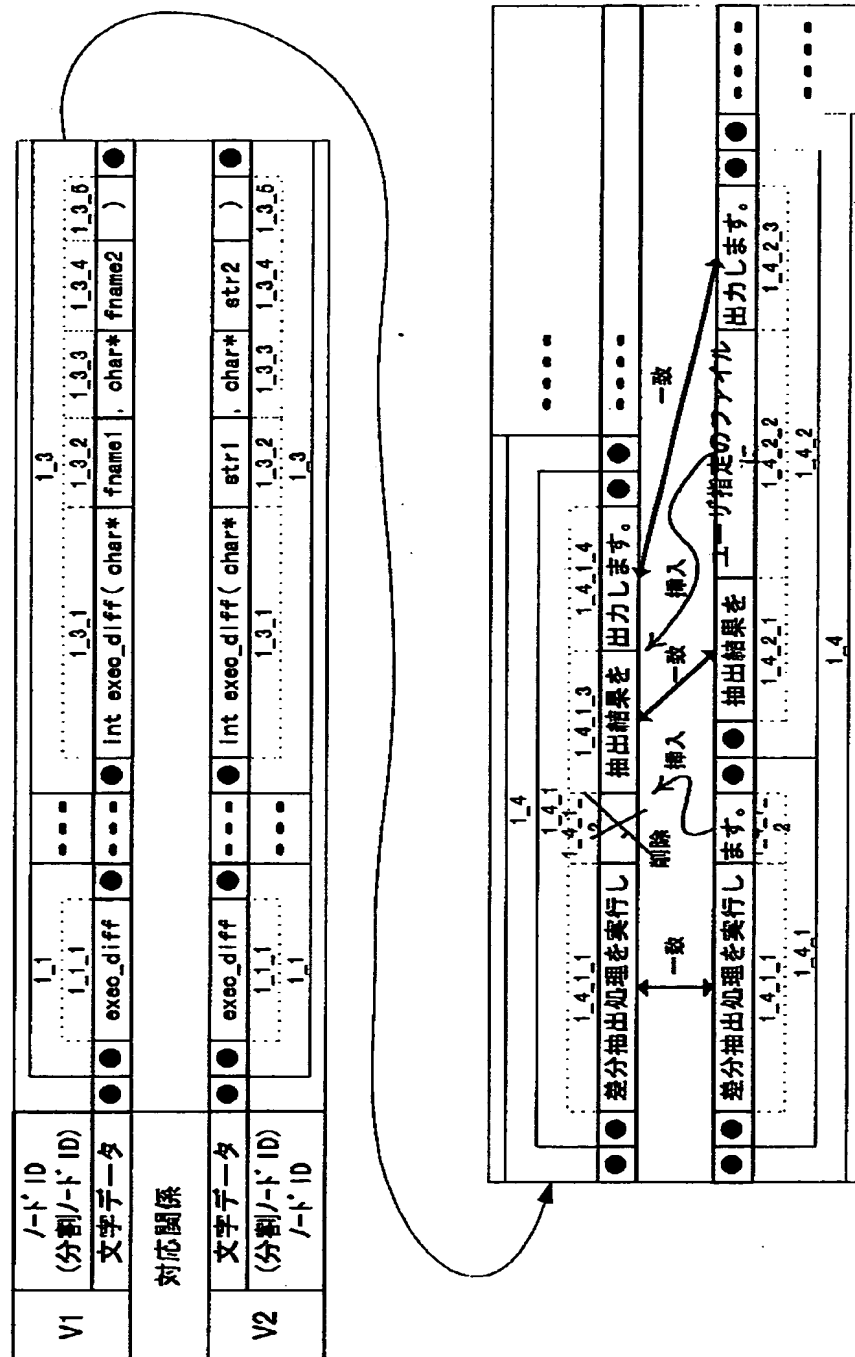
【図 7】

図 7

文字No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14															41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
V1	●	●	e	x	e	c	-	d	i	f	f	●	●	差		h	a	r	*		f	n	a	m	e	i	.		o	h	a	r													
V2	●	●	e	x	e	c	-	d	i	f	f	●	●	差		h	a	r	*		s	t	r	i	.		o	h	a	r	*														

【圖 8】



【図 9】

図 9

ノード ID	分割ノード ID	文字 No.	共通/差分	差分タイプ	内容文字列	対応分割ノード ID	対応ノード ID
1_1	1_1_1	3~11	共通	-	exec diff	1_1_1	1_1
1_2	1_2_1	14~22	共通	-	差分抽出処理の実行	1_2_1	1_2
1_3	1_3_1	25~45	共通	-	int exec_diff(char*	1_3_1	1_3
1_3	1_3_2	48~51	差分	変更	fname1	1_3_2	1_3
1_3	1_3_3	52~59	共通	-	, char*	1_3_3	1_3
1_3	1_3_4	60~65	差分	変更	fname2	1_3_4	1_3
1_3	1_3_5	66~67	共通	-)	1_3_5	1_3
1_4_1	1_4_1_1	71~80	共通	-	差分抽出処理を実行し	1_4_1_1	1_4_1
1_4_1	1_4_1_2	81	差分	削除		-	-
1_4_1	1_4_1_3	82~88	共通	-	抽出結果を	1_4_2_1	1_4_2
1_4_1	1_4_1_4	87~92	共通	-	出力します。	1_4_2_3	1_4_2
...							

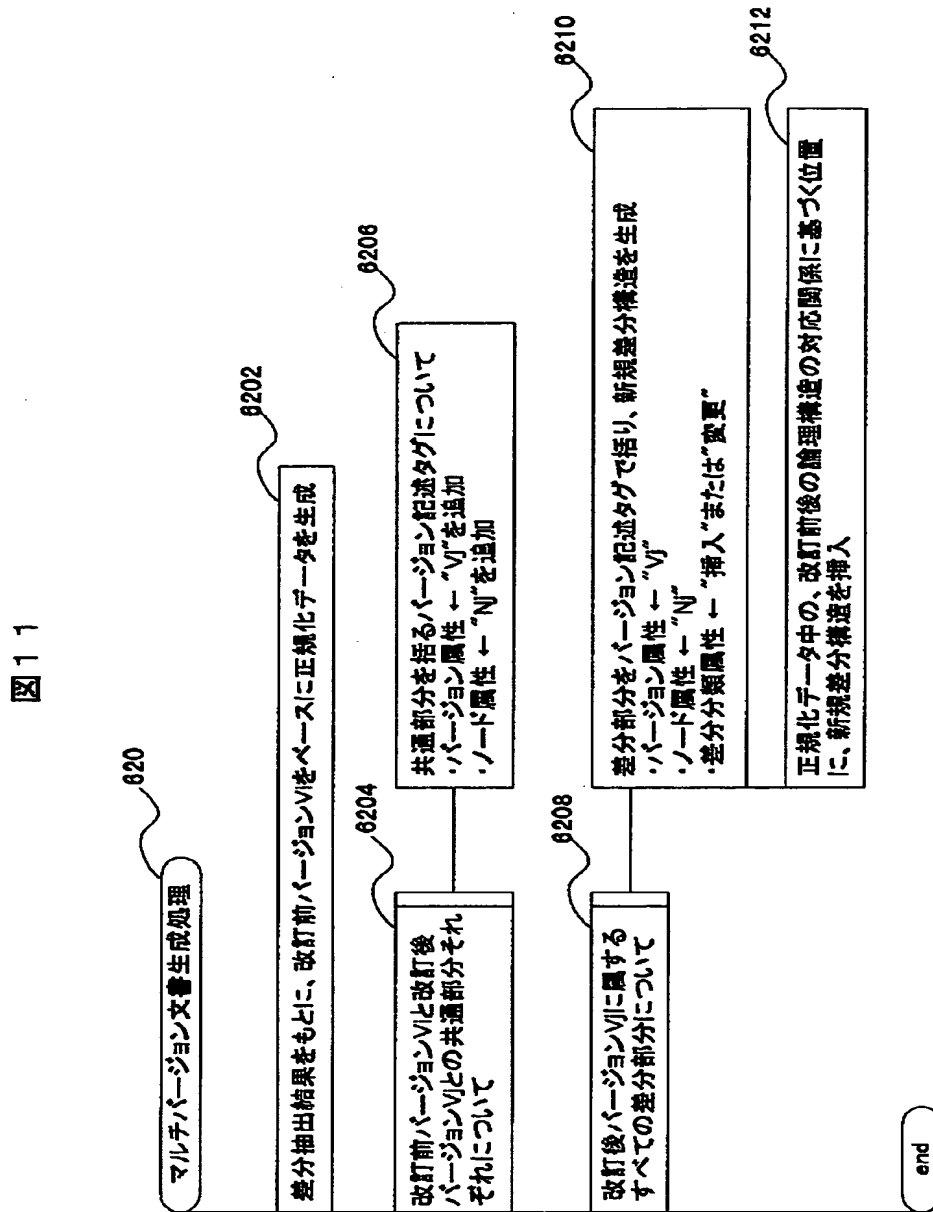
【図 10】

図 10

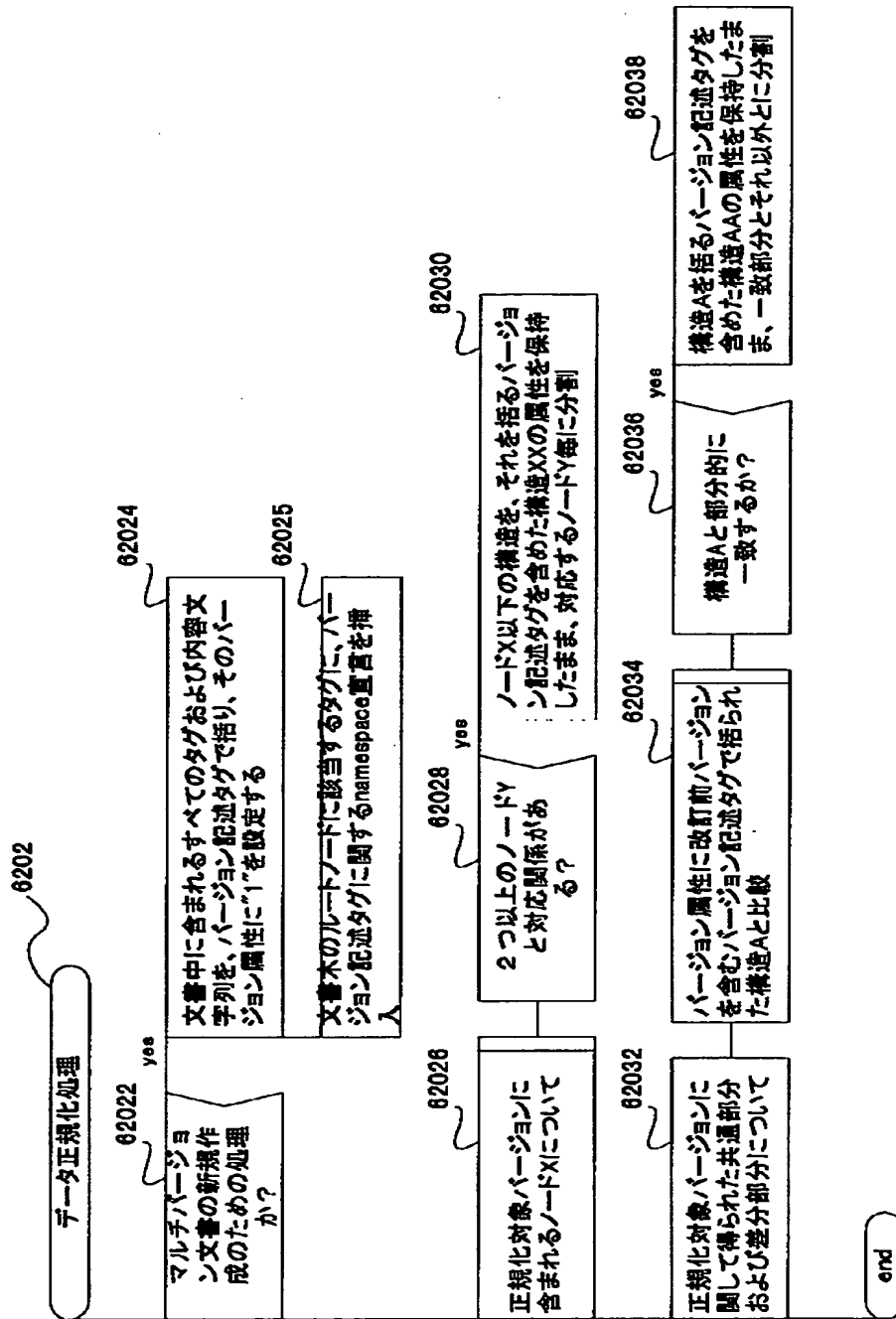
ノード ID	分割ノード ID	文字 No.	共通/差分	差分タイプ	内容文字列	対応分割ノード ID	対応ノード ID
1_1	1_1_1	3~11	共通	—	exeo_diff	1_1_1	1_1
1_2	1_2_1	14~22	共通	—	差分抽出処理の実行	1_2_1	1_2
1_3	1_3_1	25~45	共通	—	int exeo_diff(char*)	1_3_1	1_3
1_3	1_3_2	46~49	差分	変更	str1	1_3_2	1_3
1_3	1_3_3	50~57	共通	—	, char*	1_3_3	1_3
1_3	1_3_4	58~61	差分	変更	str2	1_3_4	1_3
1_3	1_3_5	62~63	共通	—)	1_3_5	1_3
1_4_1	1_4_1_1	67~76	共通	—	差分抽出処理を実行します。	1_4_1_1	1_4_1
1_4_1	1_4_1_2	77~79	差分	挿入	抽出結果を	—	—
1_4_2	1_4_2_1	82~86	共通	—	ユーザ指定のファイルに	1_4_1_3	1_4_1
1_4_2	1_4_2_2	87~97	差分	挿入	出力します。	—	—
1_4_2	1_4_2_3	98~103	共通	—		1_4_1_4	1_4_1
!							

67-2

【図 1 1】



【図 1 2】



【図 13】

図 13

```

<関数一覧 xmlns:diff='http://www.xxx.yyy/diff/'>
...
<diff:ver vnum="1" nodeID="1"><関数情報>
  <diff:ver vnum="1" nodeID="1.1"><関数名称>
    <diff:ver vnum="1" nodeID="1.1.1">exec_diff</diff:ver></関数名称></diff:ver>
  <diff:ver vnum="1" nodeID="1.2"><機能>
    <diff:ver vnum="1" nodeID="1.2.1">差分抽出処理の実行</diff:ver></機能></diff:ver>
  <diff:ver vnum="1" nodeID="1.3"><形式>
    <diff:ver vnum="1" nodeID="1.3.1">int exec_diff( char* </diff:ver>
    <diff:ver vnum="1" nodeID="1.3.2">fname1</diff:ver>
    <diff:ver vnum="1" nodeID="1.3.3">, char* </diff:ver>
    <diff:ver vnum="1" nodeID="1.3.4">fname2</diff:ver>
    <diff:ver vnum="1" nodeID="1.3.5"> )</diff:ver></形式></diff:ver>
  <diff:ver vnum="1" nodeID="1.4"><説明>
    <diff:ver vnum="1" nodeID="1.4.1"><段落>
      <diff:ver vnum="1" nodeID="1.4.1.1">差分抽出処理を実行し</diff:ver>
      <diff:ver vnum="1" nodeID="1.4.1.2">.</diff:ver></段落></diff:ver>
    <diff:ver vnum="1" nodeID="1.4.1"><段落>
      <diff:ver vnum="1" nodeID="1.4.1.3">抽出結果を</diff:ver>
      <diff:ver vnum="1" nodeID="1.4.1.4">出力します.</diff:ver></段落></diff:ver>
    </説明></diff:ver>
  <diff:ver vnum="1" nodeID="1.5"><引数リスト>
    <diff:ver vnum="1" nodeID="1.5.1"><引数>
      <diff:ver vnum="1" nodeID="1.5.1.1"><名称>
        <diff:ver vnum="1" nodeID="1.5.1.1.1">fname1</diff:ver></名称></diff:ver>
      <diff:ver vnum="1" nodeID="1.5.1.2"><内容>
        <diff:ver vnum="1" nodeID="1.5.1.2.1">比較対象</diff:ver>
        <diff:ver vnum="1" nodeID="1.5.1.2.2">ファイル名</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    <diff:ver vnum="1" nodeID="1.5.2"><引数>
      <diff:ver vnum="1" nodeID="1.5.2.1"><名称>
        <diff:ver vnum="1" nodeID="1.5.2.1.1">fname2</diff:ver></名称></diff:ver>
      <diff:ver vnum="1" nodeID="1.5.2.2"><内容>
        <diff:ver vnum="1" nodeID="1.5.2.2.1">比較対象</diff:ver>
        <diff:ver vnum="1" nodeID="1.5.2.2.2">ファイル名</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    </引数リスト></diff:ver>
  <diff:ver vnum="1" nodeID="1.6"><戻り値リスト>
    <diff:ver vnum="1" nodeID="1.6.1"><戻り値>
      <diff:ver vnum="1" nodeID="1.6.1.1"><値>
        <diff:ver vnum="1" nodeID="1.6.1.1.1">0</diff:ver></値></diff:ver>
      <diff:ver vnum="1" nodeID="1.6.1.2"><詳細>
        <diff:ver vnum="1" nodeID="1.6.1.2.1">正常終了</diff:ver></詳細></diff:ver>
      </戻り値></diff:ver>
    </戻り値リスト></diff:ver>
  <diff:ver vnum="1" nodeID="1.7"><注意事項>
    <diff:ver vnum="1" nodeID="1.7.1">特になし</diff:ver></注意事項></diff:ver>
</関数情報></diff:ver>
...
</関数一覧>

```

【図14】

図14

```

<関数一覧 xmlns:diff='http://www.xxx.yyy/diff/'>
...
<diff:ver vnum="1 2" nodeId="1 1"><関数情報>
  <diff:ver vnum="1 2" nodeId="1 1 1 1"><関数名称>
    <diff:ver vnum="1 2" nodeId="1 1 1 1 1">exec_diff</diff:ver></関数名称></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 2 1 2"><機能>
    <diff:ver vnum="1 2" nodeId="1 2 1 1 2 1">差分抽出処理の実行</diff:ver></機能></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 3 1 3"><形式>
    <diff:ver vnum="1 2" nodeId="1 3 1 1 3 1">int exec_diff( char* </diff:ver>
    <diff:ver vnum="1" nodeId="1 3 2">fname1</diff:ver>
    <diff:ver vnum="2" nodeId="1 3 2" desc="変更">str1</diff:ver>
    <diff:ver vnum="1 2" nodeId="1 3 3 1 3 3">, char* </diff:ver>
    <diff:ver vnum="1" nodeId="1 3 4">fname2</diff:ver>
    <diff:ver vnum="2" nodeId="1 3 4" desc="変更">str2</diff:ver>
    <diff:ver vnum="1 2" nodeId="1 3 5 1 3 5"> )</diff:ver></形式></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 4 1 4"><説明>
    <diff:ver vnum="1 2" nodeId="1 4 1 1 4 1"><段落>
      <diff:ver vnum="1 2" nodeId="1 4 1 1 1 4 1 1">差分抽出処理を実行し</diff:ver>
      <diff:ver vnum="1" nodeId="1 4 1 2">, </diff:ver>
      <diff:ver vnum="2" nodeId="1 4 1 2" desc="挿入">ます。</diff:ver></段落></diff:ver>
    <diff:ver vnum="1 2" nodeId="1 4 1 1 4 2"><段落>
      <diff:ver vnum="1 2" nodeId="1 4 1 3 1 4 2 1">抽出結果を</diff:ver>
      <diff:ver vnum="2" nodeId="1 4 2 2" desc="挿入">ユーザ指定のファイルに</diff:ver>
      <diff:ver vnum="1 2" nodeId="1 4 1 4 1 4 2 3">出力します。</diff:ver></段落></diff:ver>
    </説明></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 5 1 5"><引数リスト>
    <diff:ver vnum="1 2" nodeId="1 5 1 1 5 1"><引数>
      <diff:ver vnum="1 2" nodeId="1 5 1 1 1 5 1 1"><名称>
        <diff:ver vnum="1" nodeId="1 5 1 1 1">fname1</diff:ver>
        <diff:ver vnum="2" nodeId="1 5 1 1 1" desc="変更">str1</diff:ver></名称></diff:ver>
      <diff:ver vnum="1 2" nodeId="1 5 1 2 1 5 1 2"><内容>
        <diff:ver vnum="1 2" nodeId="1 5 1 2 1 1 5 1 2 1">比較対象</diff:ver>
        <diff:ver vnum="1" nodeId="1 5 1 2 2">ファイル名</diff:ver>
        <diff:ver vnum="2" nodeId="1 5 1 2 2" desc="挿入">文字列</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    <diff:ver vnum="1 2" nodeId="1 5 2 1 5 2"><引数>
      <diff:ver vnum="1 2" nodeId="1 5 2 1 1 5 2 1"><名称>
        <diff:ver vnum="1" nodeId="1 5 2 1 1">fname2</diff:ver>
        <diff:ver vnum="2" nodeId="1 5 2 1 1" desc="変更">str2</diff:ver></名称></diff:ver>
      <diff:ver vnum="1 2" nodeId="1 5 2 2 1 5 2 2"><内容>
        <diff:ver vnum="1 2" nodeId="1 5 2 2 1 1 5 2 2 1">比較対象</diff:ver>
        <diff:ver vnum="1" nodeId="1 5 2 2 2">ファイル名</diff:ver>
        <diff:ver vnum="2" nodeId="1 5 2 2 2" desc="挿入">文字列</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    </引数リスト></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 6 1 6"><戻り値リスト>
    <diff:ver vnum="1 2" nodeId="1 6 1 1 6 1"><戻り値>
      <diff:ver vnum="1 2" nodeId="1 6 1 1 1 6 1 1"><値>
        <diff:ver vnum="1 2" nodeId="1 6 1 1 1 1 6 1 1 1">0</diff:ver></値></diff:ver>
      <diff:ver vnum="1 2" nodeId="1 6 1 2 1 6 1 2"><詳細>
        <diff:ver vnum="1 2" nodeId="1 6 1 2 1 1 6 1 2 1">正常終了</diff:ver></詳細></diff:ver>
    ...

```

【図15】

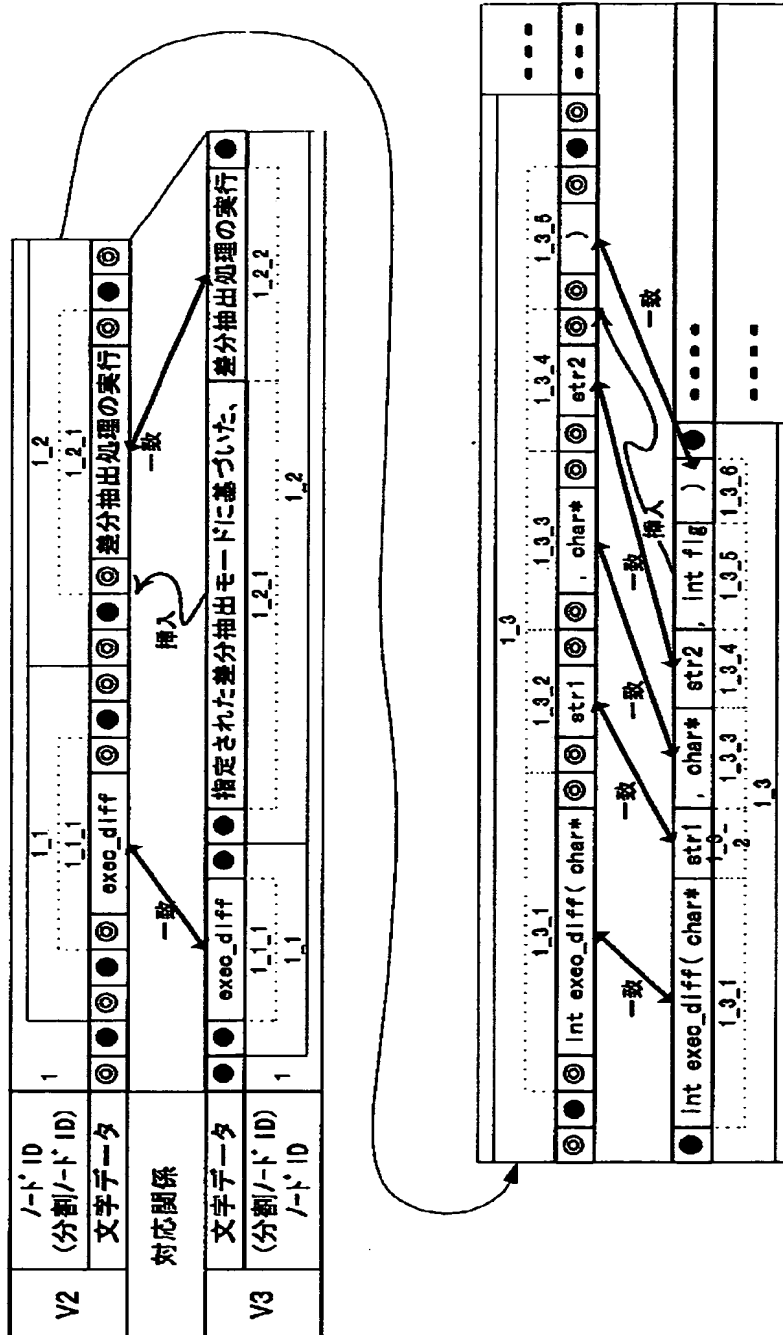
図15

```

<関数一覧 xmlns:diff='http://www.xxx.yyy/diff/'>
...
<diff:ver vnum="1 2" nodeId="1 1"><関数情報>
  <diff:ver vnum="1 2" nodeId="1 1 1 1"><関数名>
    <diff:ver vnum="1 2" nodeId="1 1 1 1 1">exec_diff</diff:ver></関数名></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 2 1 2"><機能>
    <diff:ver vnum="1 2" nodeId="1 2 1 1 2 1">差分抽出処理の実行</diff:ver></機能></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 3 1 3"><形式>
    <diff:ver vnum="1 2" nodeId="1 3 1 1 3 1">int exec_diff( char* </diff:ver>
    <diff:ver vnum="2" nodeId="1 3 2" desc="変更">str1</diff:ver>
    <diff:ver vnum="1 2" nodeId="1 3 3 1 3 3">. char* </diff:ver>
    <diff:ver vnum="2" nodeId="1 3 4" desc="変更">str2</diff:ver>
    <diff:ver vnum="1 2" nodeId="1 3 5 1 3 5"> )</diff:ver></形式></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 4 1 4"><説明>
    <diff:ver vnum="1 2" nodeId="1 4 1 1 4 1"><段落>
      <diff:ver vnum="1 2" nodeId="1 4 1 1 1 4 1 1">差分抽出処理を実行し</diff:ver>
      <diff:ver vnum="2" nodeId="1 4 1 2" desc="挿入">ます。</diff:ver></段落></diff:ver>
    <diff:ver vnum="1 2" nodeId="1 4 1 1 4 2"><段落>
      <diff:ver vnum="1 2" nodeId="1 4 1 3 1 4 2 1">抽出結果を</diff:ver>
      <diff:ver vnum="2" nodeId="1 4 2 2" desc="挿入">ユーザ指定のファイルに</diff:ver>
      <diff:ver vnum="1 2" nodeId="1 4 1 4 1 4 2 3">出力します。</diff:ver></段落></diff:ver>
    </説明></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 5 1 5"><引数リスト>
    <diff:ver vnum="1 2" nodeId="1 5 1 1 5 1"><引数>
      <diff:ver vnum="1 2" nodeId="1 5 1 1 1 5 1 1"><名称>
        <diff:ver vnum="2" nodeId="1 5 1 1 1" desc="変更">str1</diff:ver></名称></diff:ver>
      <diff:ver vnum="1 2" nodeId="1 5 1 2 1 5 1 2"><内容>
        <diff:ver vnum="1 2" nodeId="1 5 1 2 1 1 5 1 2 1">比較対象</diff:ver>
        <diff:ver vnum="2" nodeId="1 5 1 2 2" desc="挿入">文字列</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    <diff:ver vnum="1 2" nodeId="1 5 2 1 5 2"><引数>
      <diff:ver vnum="1 2" nodeId="1 5 2 1 1 5 2 1"><名称>
        <diff:ver vnum="2" nodeId="1 5 2 1 1" desc="変更">str2</diff:ver></名称></diff:ver>
      <diff:ver vnum="1 2" nodeId="1 5 2 2 1 5 2 2"><内容>
        <diff:ver vnum="1 2" nodeId="1 5 2 2 1 1 5 2 2 1">比較対象</diff:ver>
        <diff:ver vnum="2" nodeId="1 5 2 2 2" desc="挿入">文字列</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    </引数リスト></diff:ver>
  <diff:ver vnum="1 2" nodeId="1 6 1 6"><戻り値リスト>
    <diff:ver vnum="1 2" nodeId="1 6 1 1 6 1"><戻り値>
      <diff:ver vnum="1 2" nodeId="1 6 1 1 1 6 1 1"><値>
        <diff:ver vnum="1 2" nodeId="1 6 1 1 1 1 6 1 1 1">0</diff:ver></値></diff:ver>
      <diff:ver vnum="1 2" nodeId="1 6 1 2 1 6 1 2"><詳細>
        <diff:ver vnum="1 2" nodeId="1 6 1 2 1 1 6 1 2 1">正常終了</diff:ver></詳細>
      </戻り値></diff:ver>
    <diff:ver vnum="2" nodeId="1 6 2" desc="挿入"><戻り値>
      <diff:ver vnum="2" nodeId="1 6 2 1" desc="挿入"><値>
        <diff:ver vnum="2" nodeId="1 6 2 1 1" desc="挿入">ERR_1</diff:ver></値></diff:ver>
      <diff:ver vnum="2" nodeId="1 6 2 2" desc="挿入"><詳細>
        <diff:ver vnum="2" nodeId="1 6 2 2 1" desc="挿入">引数が不正です。</diff:ver>
      </戻り値></diff:ver>
    ...
  ...

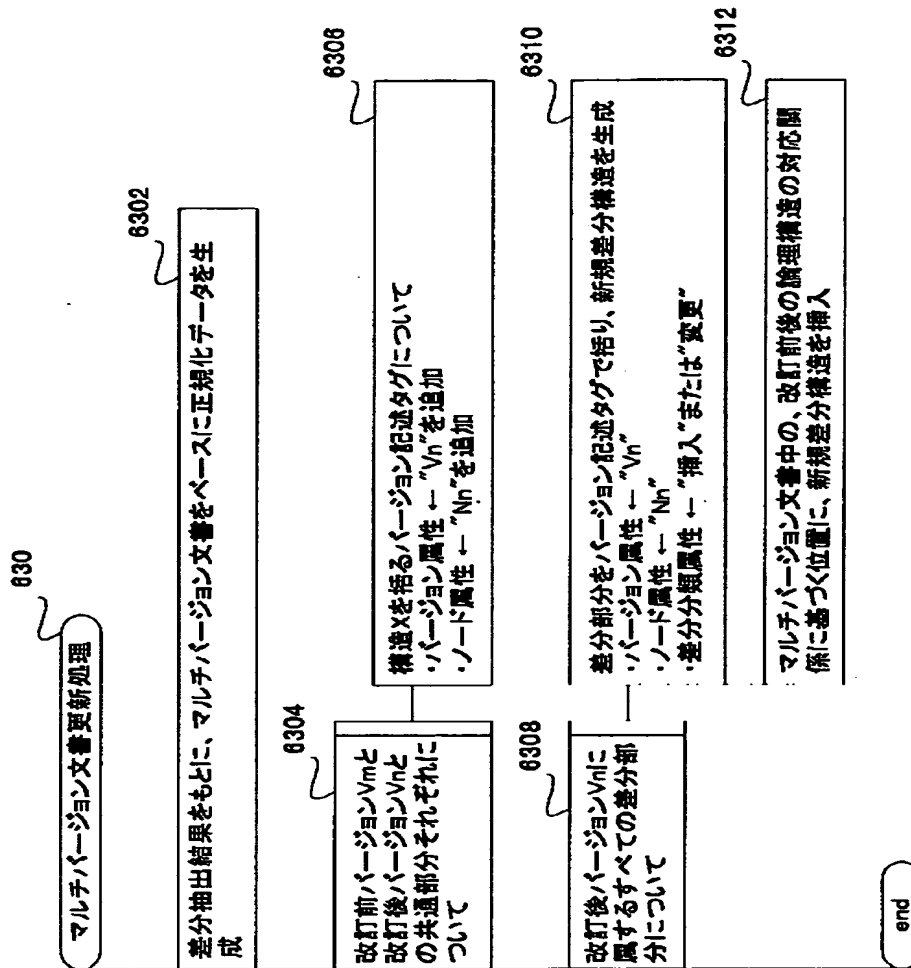
```

【図 16】



【図 17】

図 17



【図 18】

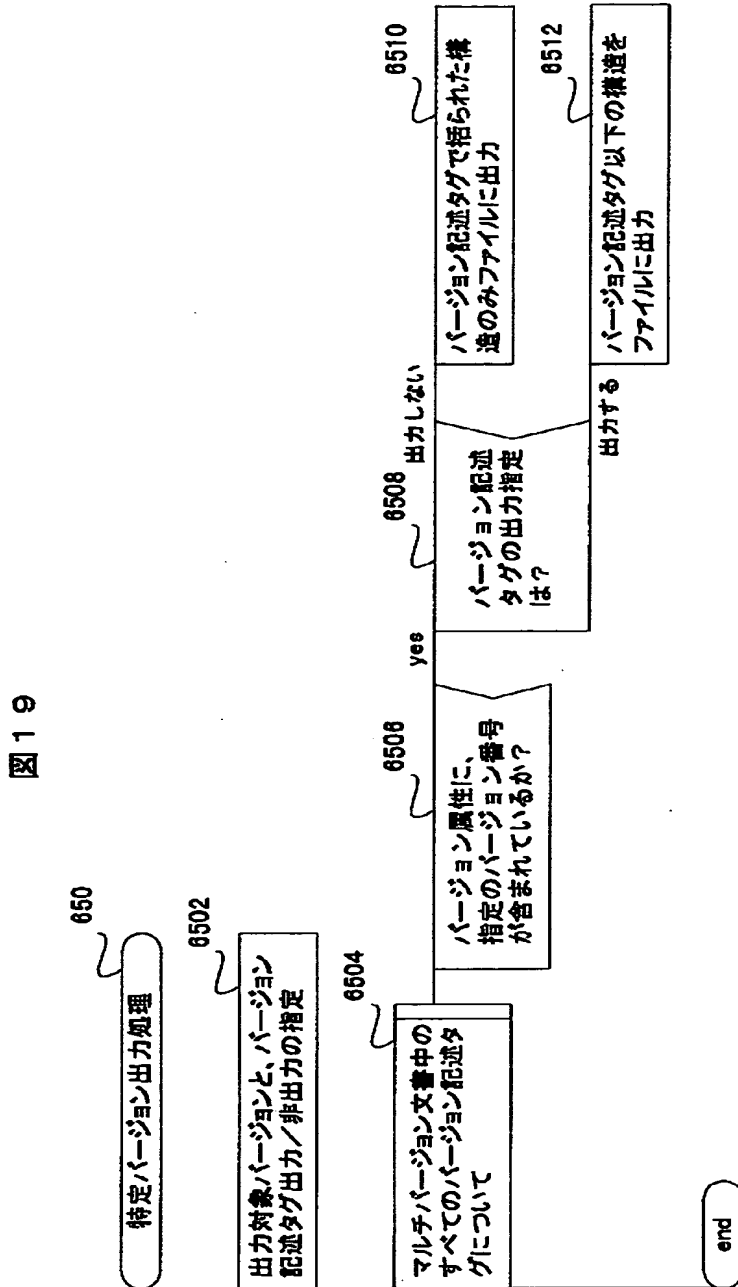
図 18

```

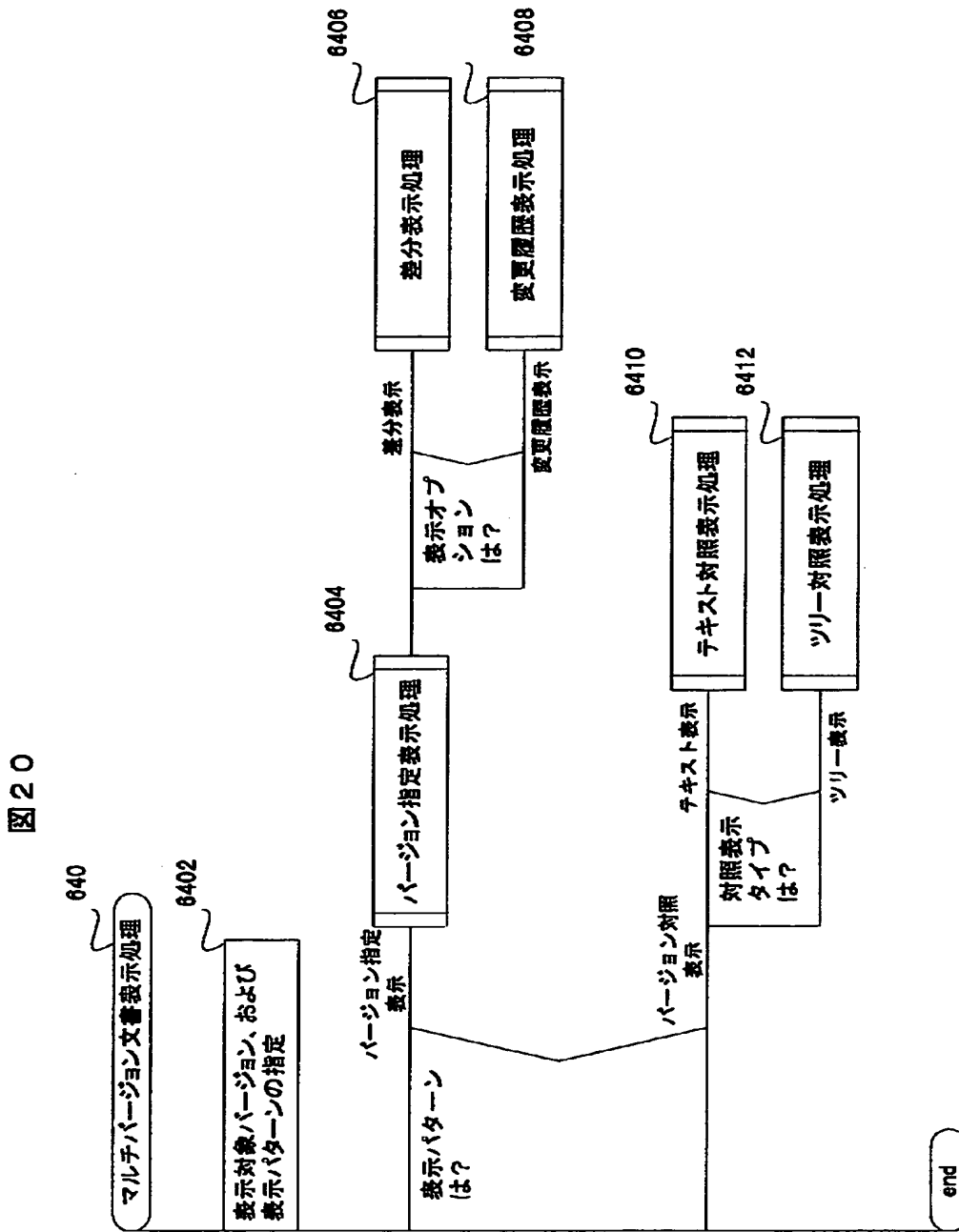
<関数一覧 xmlns:diff='http://www.xxx.yyy/diff/'>
...
<diff:ver vnum="1 2 3" nodeId="1 1 1"><関数情報>
  <diff:ver vnum="1 2 3" nodeId="1 1 1 1 1"><関数名>
    <diff:ver vnum="1 2 3" nodeId="1 1 1 1 1 1 1">exec_diff</diff:ver></関数名></diff:ver>
  <diff:ver vnum="1 2 3" nodeId="1 2 1 2 2"><機能>
    <diff:ver vnum="3" nodeId="1 2 1">指定された差分抽出モードに基づいた、</diff:ver>
    <diff:ver vnum="1 2 3" nodeId="1 2 1 1 2 1 1 2 2">差分抽出処理の実行</diff:ver>
  </機能></diff:ver>
  <diff:ver vnum="1 2 3" nodeId="1 3 1 3 1 3"><形式>
    <diff:ver vnum="1 2 3" nodeId="1 3 1 1 3 1 1 3 1">int exec_diff( char* </diff:ver>
    <diff:ver vnum="1" nodeId="1 3 2">fname1</diff:ver>
    <diff:ver vnum="2 3" nodeId="1 3 2 1 3 2">str1</diff:ver>
    <diff:ver vnum="1 2 3" nodeId="1 3 3 1 3 3 1 3 3">, char* </diff:ver>
    <diff:ver vnum="1" nodeId="1 3 4">fname2</diff:ver>
    <diff:ver vnum="2 3" nodeId="1 3 4 1 3 4">str2</diff:ver>
    <diff:ver vnum="3" nodeId="1 3 5">, int flg</diff:ver>
    <diff:ver vnum="1 2 3" nodeId="1 3 5 1 3 5 1 3 6"> )</diff:ver></形式></diff:ver>
  <diff:ver vnum="1 2 3" nodeId="1 4 1 4 1 4"><説明>
    <diff:ver vnum="1 2 3" nodeId="1 4 1 1 4 1 1 4 1 1 4 1"><段落>
      <diff:ver vnum="3" nodeId="1 4 1 1">指定された差分抽出モードに基づき、</diff:ver>
      <diff:ver vnum="1 2 3" nodeId="1 4 1 1 1 4 1 1 1 4 1 2">差分抽出処理を実行し</diff:ver>
      <diff:ver vnum="1" nodeId="1 4 1 2">、</diff:ver>
      <diff:ver vnum="2 3" nodeId="1 4 1 2 1 4 1 3">ます。</diff:ver></段落></diff:ver>
    <diff:ver vnum="3" nodeId="1 4 2"><段落>
      <diff:ver vnum="3" nodeId="1 4 2 1">差分抽出モードには、一般モードと詳細モードがあります。
      </diff:ver></段落></diff:ver>
    <diff:ver vnum="1 2 3" nodeId="1 4 1 1 4 2 1 4 3"><段落>
      <diff:ver vnum="1 2 3" nodeId="1 4 1 3 1 4 2 1 1 4 3 1">抽出結果を</diff:ver>
      <diff:ver vnum="2 3" nodeId="1 4 2 2 1 4 3 2">ユーザ指定のファイルに</diff:ver>
      <diff:ver vnum="1 2 3" nodeId="1 4 1 4 1 4 2 3 1 4 3 3">出力します。</diff:ver>
    </段落></diff:ver>
  </説明></diff:ver>
  <diff:ver vnum="1 2 3" nodeId="1 5 1 5 1 5"><引数リスト>
    <diff:ver vnum="1 2 3" nodeId="1 5 1 1 5 1 1 5 1"><引数>
      <diff:ver vnum="1 2 3" nodeId="1 5 1 1 1 5 1 1 1 5 1 1"><名称>
        <diff:ver vnum="1" nodeId="1 5 1 1 1">fname1</diff:ver>
        <diff:ver vnum="2 3" nodeId="1 5 1 1 1 1 5 1 1 1">str1</diff:ver></名称></diff:ver>
      <diff:ver vnum="1 2 3" nodeId="1 5 1 2 1 5 1 2 1 5 1 2"><内容>
        <diff:ver vnum="1 2 3" nodeId="1 5 1 2 1 1 5 1 2 1 1 5 1 2 1">比較対象</diff:ver>
        <diff:ver vnum="1" nodeId="1 5 1 2 2">ファイル名</diff:ver>
        <diff:ver vnum="2 3" nodeId="1 5 1 2 2 1 5 1 2 2">文字列</diff:ver></内容></diff:ver>
      </引数></diff:ver>
    <diff:ver vnum="1 2 3" nodeId="1 5 2 1 5 2 1 5 2"><引数>
      <diff:ver vnum="1 2 3" nodeId="1 5 2 1 1 5 2 1 1 5 2 1"><名称>
        <diff:ver vnum="1" nodeId="1 5 2 1 1">fname2</diff:ver>
        <diff:ver vnum="2 3" nodeId="1 5 2 1 1 1 5 2 1 1">str2</diff:ver></名称></diff:ver>
      <diff:ver vnum="1 2 3" nodeId="1 5 2 2 1 5 2 2 1 5 2 2"><内容>
        <diff:ver vnum="1 2 3" nodeId="1 5 2 2 1 1 5 2 2 1 1 5 2 2 1">比較対象</diff:ver>
        <diff:ver vnum="1" nodeId="1 5 2 2 2">ファイル名</diff:ver>
        <diff:ver vnum="2 3" nodeId="1 5 2 2 2 1 5 2 2 2">文字列</diff:ver></内容></diff:ver>
    </引数></diff:ver>
  ...

```

【図 19】



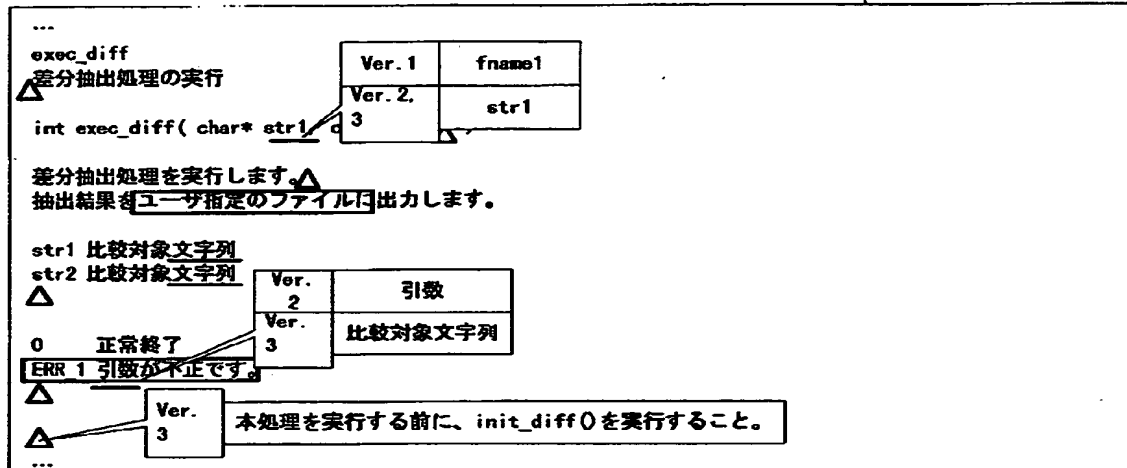
【図 20】



【図 2 1】

図 2 1

3

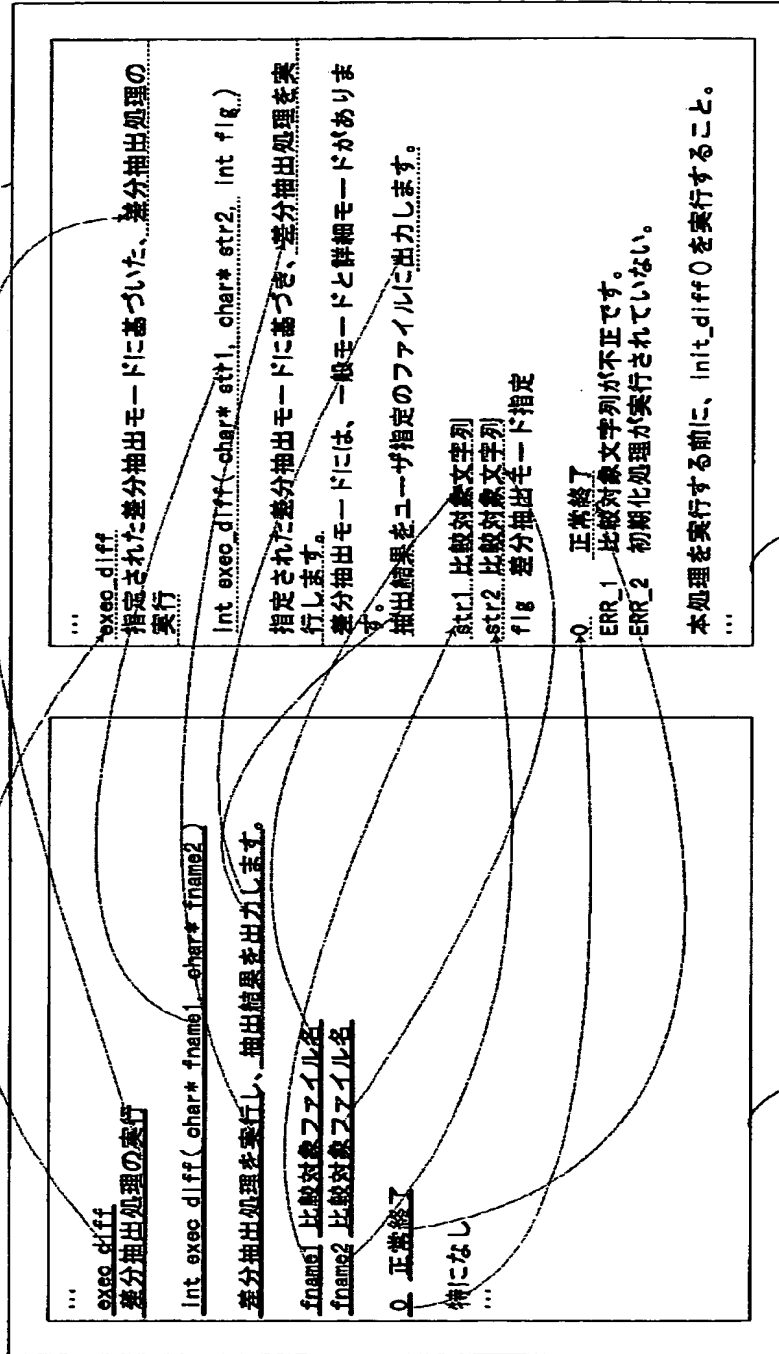


- : V2以降の変更箇所
- : V2における挿入箇所
- △ : V2よりも後の挿入箇所

【図 22】

図 22

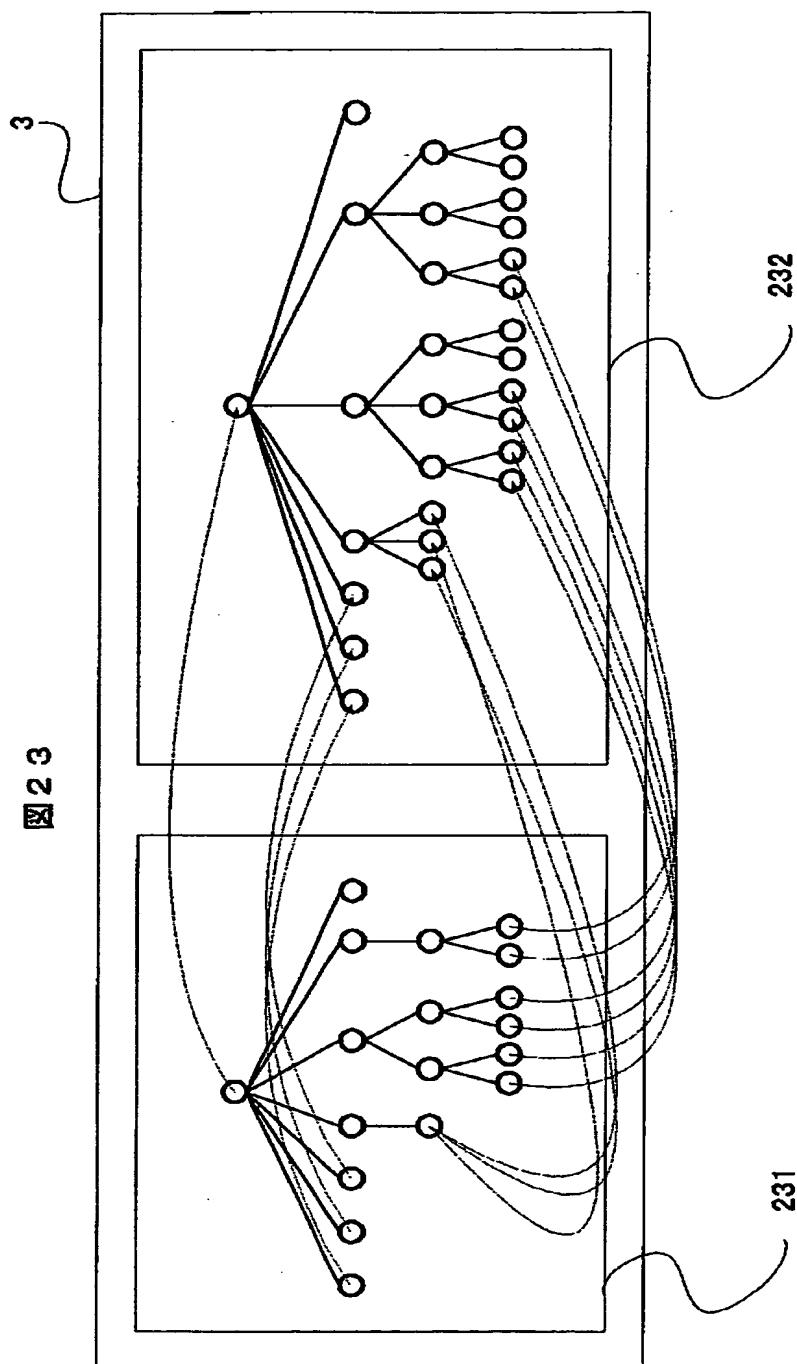
3



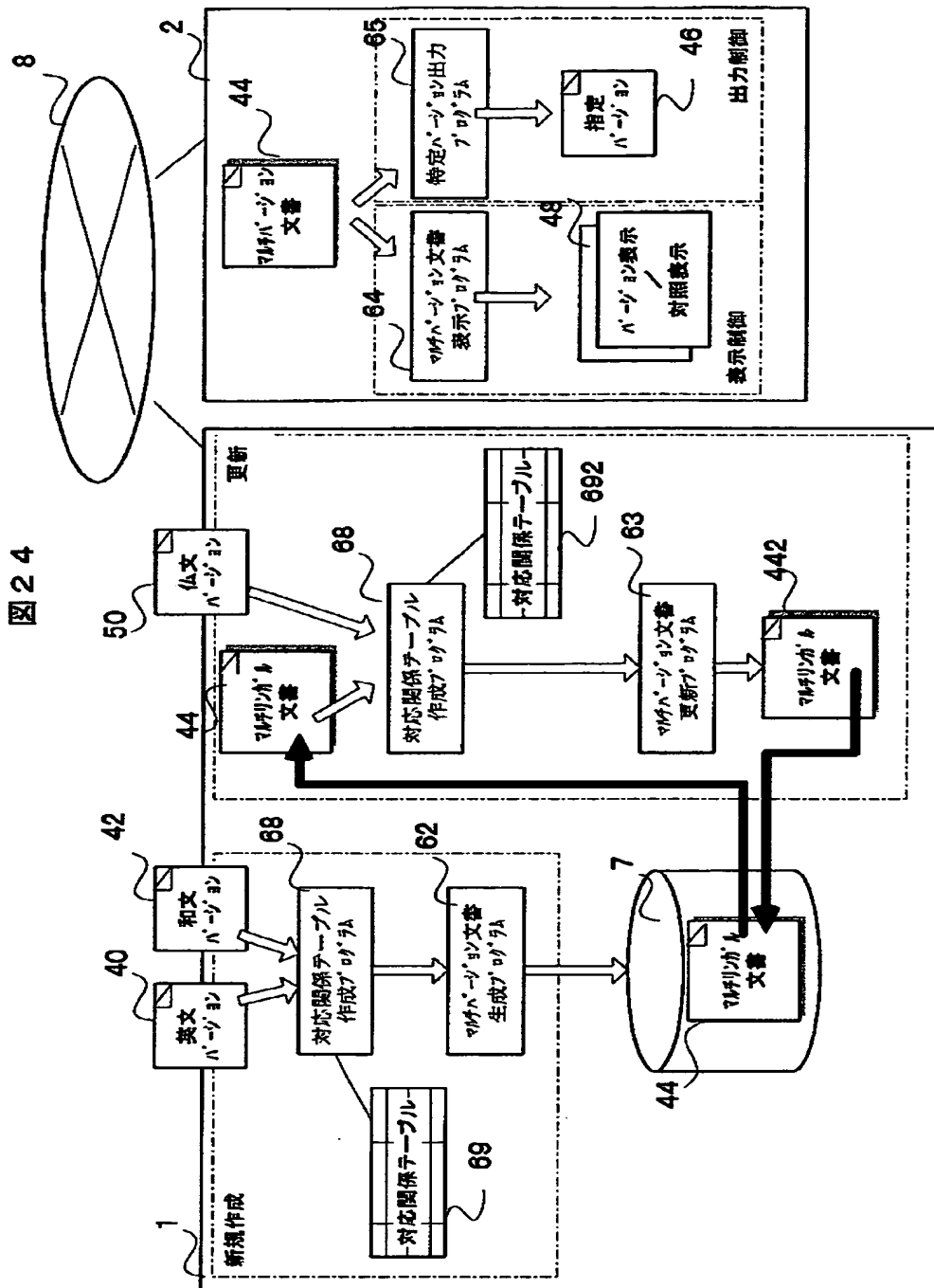
221

222

【図23】

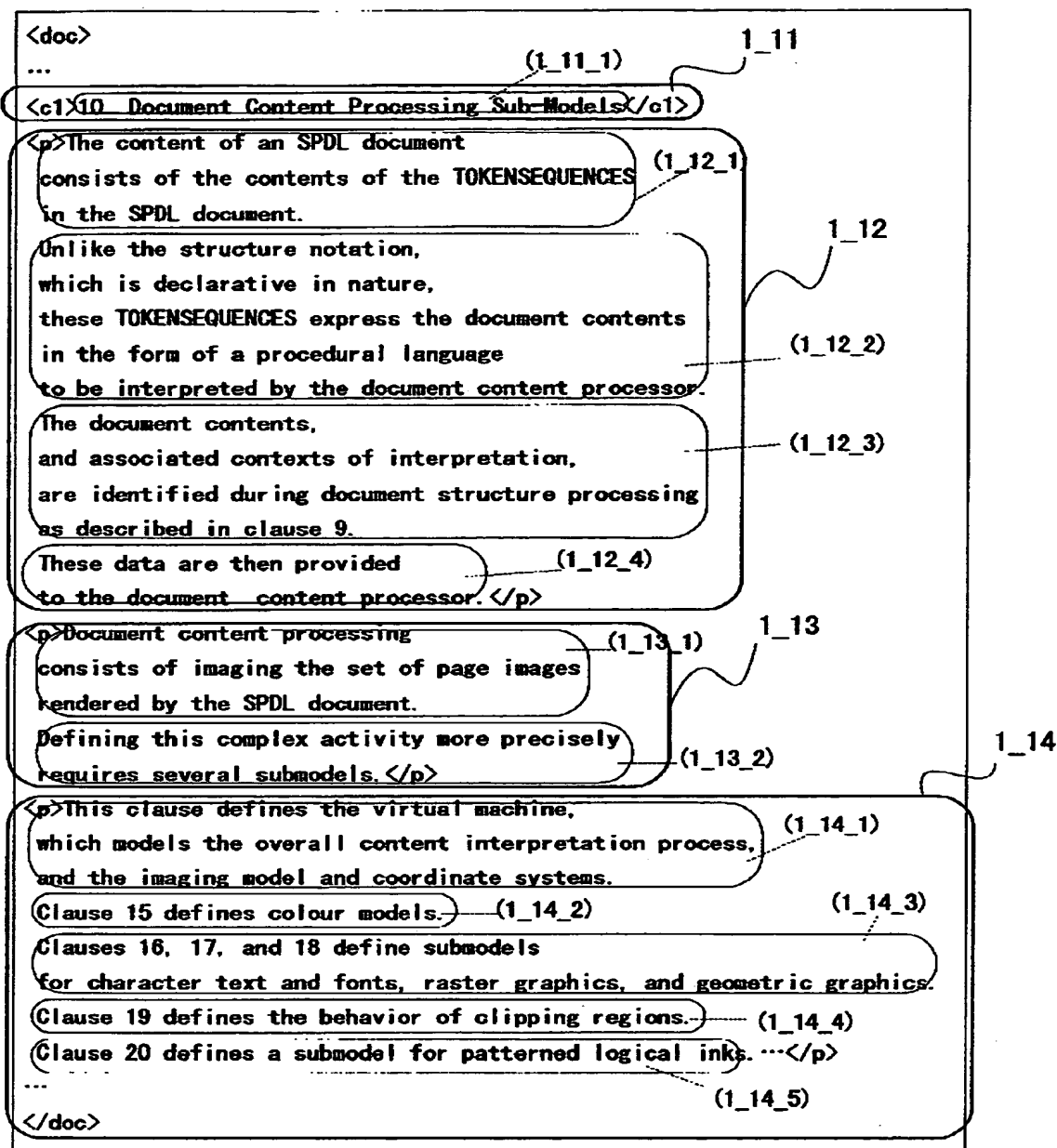


【図24】



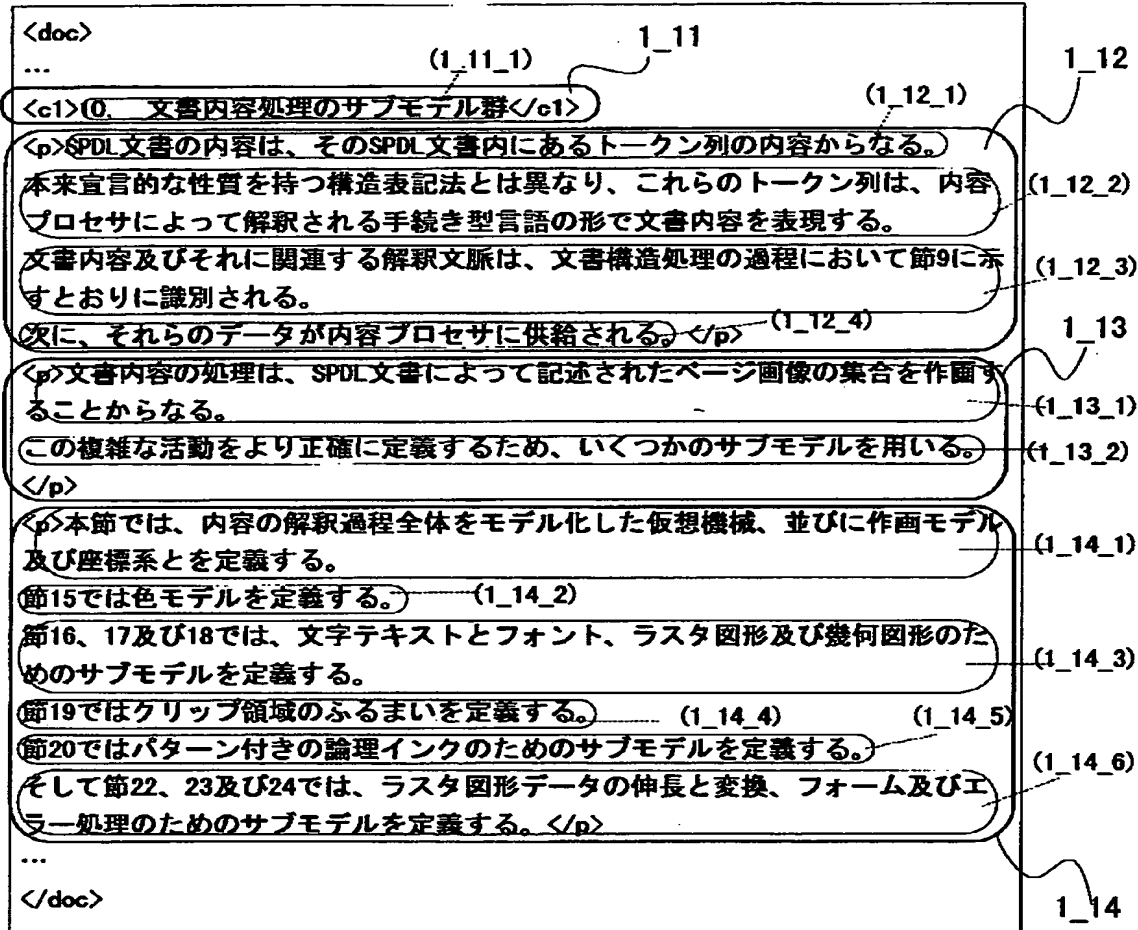
【図 25】

図 25



【図 26】

図 26



【図 2 7】

図 2 7

8911	8912	8913	8914	8915	8916	8917	8918
ノード ID	分割ノード ID	文字 No.	共通/差分	差分タイプ	内容文字列	対応分割ノード ID	対応ノード ID
1_11	1_11_1	2~42	差分	変更	10 Document Content...	1_11_1	1_11
1_12	1_12_1	45~143	差分	変更	The content of an ...	1_12_1	1_12
1_12	1_12_2	144~344	差分	変更	Unlike the structu...	1_12_2	1_12
1_12	1_12_3	345~485	差分	変更	The document content...	1_12_3	1_12
1_12	1_12_4	488~548	差分	変更	These data are then...	1_12_4	1_12
1_13	1_13_1	551~651	差分	変更	Document content pr...	1_13_1	1_13
1_13	1_13_2	652~723	差分	変更	Defining this compl...	1_13_2	1_13
1_14	1_14_1	728~866	差分	変更	This clause defines...	1_14_1	1_14
1_14	1_14_2	867~898	差分	削除	Clause 15 defines...	1_14_2	1_14
1_14	1_14_3	899~1008	差分	変更	Clause 16, 17, and 18...	1_14_3	1_14
1_14	1_14_4	1009~1059	差分	変更	Clause 19 defines...	1_14_4	1_14
;							

【図 2 8】

図 2 8

6921 ノード ID	6922 分割ノード ID	6923 文字 No.	6924 共通/差分	6925 差分タイプ	6926 内容文字列	6927 対応分割ノード ID	6928 対応ノード ID
1_11	1_11_1	2~18	差分	変更	10. 文書内容処理のサ...	1_11_1	1_11
1_12	1_12_1	21~58	差分	変更	SPDL 文書の内容は、...	1_12_1	1_12
1_12	1_12_2	57~123	差分	変更	本来宣言的な性質を...	1_12_2	1_12
1_12	1_12_3	124~170	差分	変更	文書内容及びそれに...	1_12_3	1_12
1_12	1_12_4	171~184	差分	変更	次に、それらのデ...	1_12_4	1_12
1_13	1_13_1	197~240	差分	変更	文書内容の処理は、S...	1_13_1	1_13
1_13	1_13_2	241~275	差分	変更	この複雑な活動をよ...	1_13_2	1_13
1_14	1_14_1	278~323	差分	変更	本節では、内容の解...	1_14_1	1_14
1_14	1_14_2	324~338	差分	変更	節15では色モデルを...	1_14_2	1_14
1_14	1_14_3	339~389	差分	変更	節16、17及び18では...	1_14_3	1_14
1_14	1_14_4	390~411	差分	変更	節19ではクリップ領...	1_14_4	1_14
!							

【図 29】

図 29

```

<doc xmlns:diff='http://www.xxx.yyy/diff/'>
...
<diff:ver vkind="e j">c1>
  <diff:ver vkind="e">10 Document Content Processing Sub-Models</diff:ver>
  <diff:ver vkind="j">10. 文書内容処理のサブモデル群</diff:ver>
</c1></diff:ver>
<diff:ver vkind="e j">p>
  <diff:ver vkind="e">The content of an SPDL document consists of
    the contents of the TOKENSEQUENCES in the SPDL document.</diff:ver>
  <diff:ver vkind="j">
    SPDL文書の内容は、そのSPDL文書内にあるトークン列の内容からなる。</diff:ver>
  <diff:ver vkind="e">
    Unlike the structure notation, which is declarative in nature, these
    TOKENSEQUENCES express the document contents in the form of a procedural
    language to be interpreted by the document content processor.</diff:ver>
  <diff:ver vnum="j">
    本来宣言的な性質を持つ構造表記法とは異なり、これらのトークン列は、内容プロセ
    サによって解釈される手続き型言語の形で文書内容を表現する。</diff:ver>
  <diff:ver vkind="e">The document contents, and associated contexts of
    interpretation, are identified during document structure processing as
    described in clause 9.</diff:ver>
  <diff:ver vkind="j">文書内容及びそれに関連する解釈文脈は、文書構
    造処理の過程において第9に示すとおりに識別される。</diff:ver>
  <diff:ver vkind="e">
    These data are then provided to the document content processor.</diff:ver>
  <diff:ver vkind="j">
    次に、それらのデータが内容プロセサに供給される。</diff:ver>
</p></diff:ver>
<diff:ver vkind="e j">p>
  <diff:ver vkind="e">Document content processing consists of
    imaging the set of page images rendered by the SPDL document.</diff:ver>
  <diff:ver vkind="j">文書内容の処理は、SPDL文書によって記述された
    ページ画像の集合を作画することからなる。</diff:ver>
...
</doc>

```

【書類名】 要約書

【要約】

【課題】

複数のバージョンが存在する文書について、すべてのバージョンに関する差分箇所の一括表示、および差分箇所に関する変更履歴の表示を可能とする。

【解決手段】

新たな改訂バージョンの作成時に、マルチバージョン文書に含まれる最新バージョンとの差分抽出結果に基づいて、新たな改訂バージョンの内容を上記と同様にタグ付けし、該マルチバージョン文書に足し込むマルチバージョン文書更新ステップを有することにより、複数バージョンを効率的に表現したマルチバージョン文書を生成し、このマルチバージョン文書をもとに、複数バージョンにおける変更箇所の一括表示および変更箇所に関する変更履歴表示を行う。

【選択図】 図 2

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日
[変更理由] 新規登録
住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所